•国家级特色专业（物联网工程）规划教材•

物联网技术综合实训教程

王志良　姚红串　霍磊　付洪威 ◎ 等编著

机械工业出版社
CHINA MACHINE PRESS

国家级特色专业（物联网工程）规划教材

物联网技术综合实训教程

王志良　姚红串　霍　磊　付洪威　等编著

机 械 工 业 出 版 社

本书详细介绍了物联网在智能家居领域的多个实训案例。本书共12章，由三部分组成：第一部分是智能家居总体设计；第二部分是智能家居的多个子系统实训案例，包括家庭网站系统、视频安防监控系统、Kinect体感交互系统、智能家居之RFID系统、智能家居之服务机器人、虚拟管家、移动终端之安卓开发、移动终端之Win8平板开发、健康物联网、智能家居网关系统等；第三部分摘录了已经实际参与了综合实训的学生的心得。

本书可作为高等院校物联网、计算机、电子、自动化等专业的实训课程（32学时）教材，供需要的本科生学习和研究生选读，还可作为希望了解物联网在智能家居领域的发展和应用的企业管理者、科研人员、高等院校教师等读者朋友的参考用书。

图书在版编目（CIP）数据

物联网技术综合实训教程/王志良等编著. —北京：机械工业出版社，2014.3

国家级特色专业（物联网工程）规划教材

ISBN 978-7-111-45808-1

Ⅰ.①物… Ⅱ.①王… Ⅲ.①互联网络－应用－高等学校－教材②智能技术－应用－高等学校－教材 Ⅳ.①TP393.4②TP18

中国版本图书馆CIP数据核字（2014）第026099号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）

策划编辑：朱 林 责任编辑：朱 林

版式设计：常天培 责任校对：闫玥红

封面设计：赵颖喆 责任印制：刘 岚

北京中兴印刷有限公司印刷

2014年4月第1版第1次印刷

184mm×260mm · 13.75印张 · 339千字

0 001—3 000册

标准书号：ISBN 978-7-111-45808-1

ISBN 978-7-89405-282-7（光盘）

定价：48.00元（含DVD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务 网络服务

社服务中心：(010)88361066 教 材 网：http://www.cmpedu.com

销 售 一 部：(010)68326294 机工官网：http://www.cmpbook.com

销 售 二 部：(010)88379649 机工官博：http://weibo.com/cmp1952

读者购书热线：(010)88379203 **封面无防伪标均为盗版**

前　言

物联网（Internet of Things，IOT）的发展非常迅速，它是一个基于互联网、传统电信网等信息承载体，让所有能够被独立寻址的普通物理对象实现互联互通的网络，物联网主要解决物品与物品（Thing to Thing，T2T）、人与物品（Human to Thing，H2T）、人与人（Human to Human，H2H）之间的互联。由于物联网的概念涵盖了从终端到网络、从数据采集处理到智能控制、从应用到服务、从人到物等方方面面，涉及射频识别（RFID）装置、无线传感器网络（WSN）、红外感应器、全球定位系统（GPS）、Internet 与移动网络、网络服务、行业应用软件等众多技术，是一门覆盖范围很广的综合性交叉学科。因此对于物联网工程专业的教学就提出了必须进行综合实训的基本要求，正是在这种背景下，基于这几年的物联网教学实践，按照实训实验指导书的形式，我们整理编写了此物联网技术实训教程，力求知识覆盖面大、并兼顾智能家居中多种难易程度的物联网技术，希望本书能为学生带来对物联网技术的深度感知和有效学习方面的帮助。

全书以物联网智能家居实训平台为依托按照三部分进行编写：第一部分是智能家居总体设计；第二部分是作者所开发的智能家居的多个子系统技术实训过程实例；第三部分是学生使用物联网家居综合系统实训后的心得体会。全书共分为 12 章：第 1 章智能家居总体设计，介绍了物联网和智能家居的起源、发展、市场以及智能家居实验室的总体设计、系统功能及实际构成；第 2 章家庭网站，阐述了如何搭建管理平台、设计终端软件及通过监护平台对手持终端设备的老人进行 GPS 定位、状态监测、数据分析、主动关怀等；第 3 章视频安防监控系统，通过网络摄像机对家庭进行远程实时监控，并对图像进行分析和处理；第 4 章 Kinect 体感交互实验，了解 Kinect 体感交互系统，并通过识别人体的动作来控制家电；第 5 章智能家居之 RFID 系统实验，学习 RFID 系统硬件的设计调试、RFID 系统底层硬件的编程、RFID 上位机界面程序的编写及智能家居中 RFID 系统与服务器端通信的实现；第 6 章智能家居之服务机器人实验，分硬件和软件两部分详细介绍机器人的设计和开发以及如何实现对机器人头部、行走、眼睛的控制以及唱歌、跳舞和控制家电等；第 7 章虚拟管家实验，了解智能家居网关服务系统、虚拟管家系统，学习网关服务的调用、智能家居客户端的设计、语音提示功能的实现等；第 8 章移动终端之安卓开发实验，熟悉在 Linux 环境下搭建系统，实现开发安卓手机远程监控摄像头功能；第 9 章移动终端之 Win8 平板电脑开发实验，学习如何使用 Win8 平板电脑实现智能家居电灯的控制，从而熟悉 Win8 平板电脑应用程序的开发环境和基本开发流程；第 10 章健康物联网，实验介绍了在 Win7 平台下，如何构建一个基于安卓操作系统的手机客户端软件，实现从手机客户端向服务器端提交数据申请并且返回相应的数据信息，用来跟踪身体参数变化；第 11 章智能家居网关系统，介绍了智能家居网关组成、如何配置万能学习型红外遥控器及如何对其编程来实现对智能家电的控制；第 12 章智能家居综合实训学生心得，介绍了物联网智能家居综合实训后学生的一些心得体会。

本书旨在让物联网及相关专业的学生了解掌握物联网技术在智能家居的各种应用，通过这些应用，实现理论与实践的结合。本书中的各个实验难易程度不同，每个实验涉及的知识

点不同，读者可灵活选择，且各个实验既可独立完成，又可相互联系。本书在讲述相关理论知识的同时辅以实验案例，更有利于读者对涉及的物联网各个方面技术的理解。

建议采用本书进行实训的考核具体做法如下：首先要求同学们掌握物联网家居系统整体设计（第1章）的思路；然后要求同学们选择物联网家居平台中的一个子系统（即从第2章～第11章中选择一章实训）进行软件编程和硬件配置的全程训练；最后写出实训后的心得体会。总之是希望通过学习本书内容提高同学们的物联网技术技能。建议根据读者的具体情况，将本书内容分为32学时或48学时讲授，并建议在实验室里讲解、训练本实训课程内容。

本书主要由王志良编写，王志良制定了本书大纲、内容安排并指导文字写作、负责全书的组织工作。姚红串负责全书的文字通稿工作。霍磊主持了全书各个实验的统筹工作。王志良、霍磊参与了第1章的编写工作；李爽、全力参与了第2章的编写工作；王靖元、晓虎尔参与了第3章的编写工作；付洪威、刘志强参与了第4章的编写工作；姚红串、韩振参与了第5章的编写工作；姚红串、王晨阳参与了第6章的编写工作；付洪威、霍磊参与了第7章的编写工作；于泓、谢振环参与了第8章的编写工作；于泓、闫飞参与了第9章的编写工作；李爽、王鲁参与了第10章的编写工作；王靖元、谢振环参与了第11章的编写工作；姚红串参与了第12章的编写工作。

作为全国高校物联网及其相关专业教学指导小组和物联网工程专业教学研究专家组成员，作者在其组织的物联网专业教学研讨活动中，汲取了物联网工程的教学理念，尤其是物联网专业必须加强实训的理念，作者深记心中。本书的出版得到了机械工业出版社的大力支持，在此表示诚挚的感谢。感谢国家科技重大专项项目（2010ZX07102-006）、国家973项目（2011CB505402）、国家自然科学基金项目（61170117）、省部产学研结合项目（2011A090200008）给予的支持。同时感谢国家级高等学校（物联网工程）特色专业建设点项目、北京市支持中央在京高校共建项目、北京科技大学教改重点项目的支持和资助。

本书配备有智能家居全部实验的安装使用程序，供读者选用。还提供实训过程中的技术支持，可以咨询电子邮箱：yaofeng-liu@163.com。本书实训的软、硬件环境及源代码见附带光盘。

本书引用到了互联网上的最新资讯和相关领域的最新报道等，在此一并向原作者和刊发机构致谢；由于时间、疏忽等原因对引用未能一一注明深表歉意。

由于时间仓促，加上编者水平有限，书中难免会有疏漏之处，恳请各位读者、老师批评指正，在此编者表示衷心的感谢。

编　者

2013年9月

目　录

前言

第 1 章　智能家居总体设计 …… 1

1.1　内容概览和学习目标 …… 1

1.2　物联网与智能家居 …… 2

1.3　智能家居的市场需求和意义 …… 4

1.4　智能家居实验室系统设计 …… 5

1.5　智能家居系统功能 …… 9

1.6　智能家居系统设备实物 …… 10

1.7　小结 …… 15

参考文献 …… 16

第 2 章　家庭网站 …… 17

2.1　实验目的 …… 17

2.2　实验内容 …… 17

2.3　实验所用仪表及设备 …… 17

2.3.1　系统运行环境 …… 17

2.3.2　开发技术 …… 17

2.4　实验原理 …… 18

2.5　实验程序设计 …… 19

2.5.1　代码实现 …… 19

2.5.2　界面效果 …… 21

2.5.3　数据库设计 …… 22

2.6　实验步骤 …… 24

2.6.1　环境搭建 …… 24

2.6.2　在 SQL Server Management Studio 里建立数据库 …… 26

2.6.3　在 Visual Studio 2010 里创建数据通信窗体应用程序 …… 27

2.6.4　在 Visual Studio 2010 里创建网站 …… 28

2.6.5　发布网站 …… 31

2.7　思考题 …… 33

2.8　实验报告内容及要求 …… 33

参考文献 …… 33

第 3 章　视频安防监控系统 …… 34

3.1　实验目的 …… 34

3.2　实验内容 …… 34

3.3　实验所用仪表及设备 …… 34

3.4　实验原理 …… 34

3.4.1　安防监控系统概述 …… 34

3.4.2　硬件设计与实现 …… 37

3.4.3　软件设计与实现 …… 39

3.4.4　图像处理算法 …… 40

3.5　实验步骤 …… 42

3.5.1　访问网络摄像机对家庭进行远程监控 …… 42

3.5.2　配置 VS2010 和 OpenCV …… 42

3.5.3　运动目标检测 …… 48

3.5.4　人脸检测 …… 51

3.5.5　基于阈值分割的边缘检测 …… 54

3.6　思考题 …… 56

3.7　实验报告内容及要求 …… 56

参考文献 …… 57

第 4 章　Kinect 体感交互实验 …… 58

4.1　实验目的 …… 58

4.2　实验内容 …… 58

4.3　实验所用仪表及设备 …… 58

4.4　实验原理 …… 58

4.4.1　Kinect 设备 …… 58

4.4.2　Kinect 工作原理 …… 59

4.4.3　Kinect 应用领域 …… 59

4.4.4　Kinect 在智能家居中的应用 …… 60

4.5　实验步骤 …… 61

4.5.1　Kinect 设备连接 …… 61

4.5.2　实现过程 …… 62

4.6　思考题 …… 70

4.7　实验报告内容及要求 …… 70

参考文献 …… 70

第 5 章　智能家居之 RFID 系统实验 …… 72

5.1　实验目的 …… 72

5.2　实验内容 …… 72

5.3　实验所用仪表及设备 …… 72

5.4　实验原理 …… 72

5.4.1　RFID 概述 …… 72

5.4.2 不同频段 RFID 产品特性及应用 …… 75
5.4.3 RFID 应用 …… 76
5.4.4 本实验所用的 RFID 系统 …… 77
5.5 实验步骤 …… 85
5.5.1 焊接、调试系统硬件 …… 85
5.5.2 编写 C 语言程序 …… 85
5.6 思考题 …… 97
5.7 实验报告内容及要求 …… 97
参考文献 …… 98
第 6 章 智能家居之服务机器人实验 …… 99
6.1 实验目的 …… 99
6.2 实验内容 …… 99
6.3 实验所用仪表及设备 …… 99
6.4 实验原理 …… 99
6.4.1 系统硬件环境 …… 99
6.4.2 软件环境搭建及程序设计 …… 106
6.5 实验步骤 …… 117
6.5.1 机器人运动控制 …… 117
6.5.2 机器人智能家居控制 …… 122
6.6 思考题 …… 124
6.7 实验报告内容及要求 …… 124
参考文献 …… 125
第 7 章 虚拟管家实验 …… 126
7.1 实验目的 …… 126
7.2 实验内容 …… 126
7.3 实验所用仪表及设备 …… 126
7.4 实验原理 …… 126
7.4.1 智能家居网关服务系统 …… 126
7.4.2 网关服务系统构成 …… 126
7.4.3 网关服务发布 …… 127
7.4.4 服务发布方法 …… 131
7.4.5 客户端调用网关服务 …… 132
7.4.6 虚拟管家系统 …… 132
7.4.7 语音合成服务 …… 133
7.5 实验步骤 …… 133
7.5.1 建立空白解决方案 …… 133
7.5.2 系统操作过程 …… 133
7.6 思考题 …… 148
7.7 实验报告内容及要求 …… 148
参考文献 …… 148
第 8 章 移动终端之安卓开发实验 …… 149
8.1 实验目的 …… 149
8.2 实验内容 …… 149
8.3 实验所用仪表及设备 …… 149
8.4 实验原理 …… 149
8.4.1 系统简介 …… 149
8.4.2 发展历程及发行版本 …… 149
8.4.3 系统架构 …… 150
8.4.4 平台优势 …… 152
8.5 实验程序设计 …… 152
8.5.1 程序执行流程 …… 152
8.5.2 开发环境配置 …… 153
8.5.3 设置应用程序的界面布局 Layout …… 157
8.5.4 获取远程监控画面的代码实现 …… 161
8.5.5 调试与实现 …… 168
8.6 思考题 …… 168
8.7 实验报告内容及要求 …… 168
参考文献 …… 168
第 9 章 移动终端之 Win8 平板电脑开发实验 …… 170
9.1 实验目的 …… 170
9.2 实验内容 …… 170
9.3 实验所用仪表及设备 …… 170
9.4 实验原理 …… 170
9.4.1 系统简介 …… 170
9.4.2 系统架构 …… 170
9.4.3 系统优化 …… 171
9.4.4 开发工具 …… 171
9.5 实验程序设计 …… 172
9.5.1 开发应用程序的步骤 …… 172
9.5.2 新建项目 …… 172
9.5.3 添加服务引用 …… 172
9.5.4 创建用户界面 …… 174
9.5.5 代码实现 …… 176
9.5.6 调试实现 …… 178
9.6 思考题 …… 179
9.7 实验报告内容及要求 …… 179
参考文献 …… 179
第 10 章 健康物联网 …… 180
10.1 实验目的 …… 180

10.2　实验内容 …… 180
10.3　实验所用硬件及软件 …… 180
10.3.1　硬件设备 …… 180
10.3.2　系统所用软件 …… 180
10.4　实验原理 …… 182
10.5　相关技术 …… 182
10.6　实验步骤 …… 185
10.6.1　搭建运行环境 …… 185
10.6.2　修改代码 …… 186
10.6.3　系统使用说明 …… 190
10.7　思考题 …… 194
10.8　实验报告内容及要求 …… 194
参考文献 …… 194
第 11 章　智能家居网关系统 …… 195
11.1　实验目的 …… 195
11.2　实验内容 …… 195
11.3　实验所用仪表及设备 …… 195
11.4　实验原理 …… 195
11.4.1　智能家居网关系统设计 …… 195
11.4.2　智能家居网关的硬件实现 …… 198
11.4.3　家庭网关的软件平台 …… 199
11.5　实验步骤 …… 199
11.5.1　配置学习型万能红外遥控器 …… 199
11.5.2　使用手机控制遥控器学习红外信号 …… 204
11.5.3　在 Eclipse 中模拟学习型万能红外遥控器网关指令 …… 206
11.6　思考题 …… 209
11.7　实验报告内容及要求 …… 209
参考文献 …… 209
第 12 章　智能家居综合实训学生心得 …… 210

第 1 章　智能家居总体设计

1.1　内容概览和学习目标

物联网，自从其诞生以来，已经引起了巨大关注，被认为是继计算机、互联网、移动通信网之后的又一次信息产业浪潮。目前物联网已经应用到了人们生活中的方方面面，包括智能电网、智能交通、智能物流、智能家居、金融与服务业、精细农牧业、医疗健康、环境与安全检测、国防军事等。

智能家居是物联网技术的一个分支和具体体现，也是信息类专业学生学习的一个指导性内容。目前市场上关于智能家居的相关设备已经如雨后春笋般出现。抓住智能家居这一方向学习，对于学生毕业后的发展具有巨大的意义。

本书将围绕目前物联网应用领域的热门——智能家居，对学生进行多方面的知识教导与认知实训。

本书将带领学生构建一个完整的物联网智能家居实验室平台。引进国内外先进科技，集成综合布线技术，网络通信技术，自动控制技术，音、视频技术，利用计算机和嵌入式系统，将家庭中的各种设备通过家庭网络连接到一起，建立高效的住宅设施与家庭日程事务的管理系统，提升家居安全性、便利性、舒适性、艺术性，并实现环保、节能的居住环境。

本书对智能家居的建设内容进行了全方位的分析和描述，并针对各个方面提供培训。归纳起来可以分为以下六个部分：

1）环境采集：采集生活环境的温、湿度，照度，二氧化碳浓度等信息；

2）安防监控：采集摄像头图像、红外人体传感器和烟雾传感器信息；

3）家电控制：控制灯光、电视、空调等家电设备；

4）居家健康：采集体重、血压、血氧、心率、脂肪含量等生理健康数据；

5）人机交互：利用平板电脑、智能手机、家居机器人实现多种人机交互手段；

6）智能服务：以传感器为来源、执行器为手段，分析家居状态，提供自动的家居环境和健康监护。

这些内容后面将会详细描述。

针对目前学生的学习水平和技能，本书制定了三个主要学习目标：

1）为学生学习物联网知识提供平台，破除物联网的神秘外表，使学生能够直观地认识物联网设备，并学会能够通过不同的客户端访问、操作它们，增加学生的物联网知识。

2）教导学生利用物联网技术，集成或创新使用先进的物联网设备，亲自动手，做出自己的物联网终端。这些设备将打破信息孤岛，使各个家居设备能够在家庭网络中互联，共享信息和友好操作。

3）在设备互联通的基础上，引导学生研究设备服务的内容，使数字家庭能够自主或半自主地进行自我调控，真正体现“智能”二字。

1.2 物联网与智能家居

1. 物联网的定义与发展

有关资料表明，国内、外普遍认为物联网是麻省理工学院 Ashton 教授于 1999 年最早提出来的，其理念是基于射频识别（RFID）、电子代码（EPC）等技术，在互联网的基础上，构造一个实现全球物品信息实时共享的实物互联网，即物联网。此设想有两层意思：第一，物联网的核心和基础是互联网，是在互联网基础上的延伸和扩展的网络；第二，其用户端延伸和扩展到了任何物体与物体之间，并进行信息交换和通信。

物联网技术的发展几乎涉及了信息技术的方方面面，是一种聚合性、系统性的创新应用与发展，因此被称为信息产业的第三次革命性创新。其本质主要体现在三个方面：一是互联网特征，即对需要联网的“物”一定要能够实现互联互通的网络；二是识别与通信特征，即纳入物联网的“物”一定要具备自动识别、物物通信的功能；三是智能化特征，即网络系统应具有自动化、自我反馈与智能控制的特点。

物联网为智能家居提供了技术条件，使智能家居成为可能，表现在：物联网所包括的射频技术、计算机技术、网络通信技术、综合布线技术、信息协议交换使得物品具有数据化的身份标志，借助三网融合，数据可以在电信网、互联网、广电网上对内和对外流动。智能家居是物联网技术应用生活的具体表现，使一个抽象概念转变成现实例子，表现在可实现家庭自动化、家庭网络、网络家电、信息家电等部分。

物联网对智能家居的技术贡献如表 1-1 所示。

表 1-1 物联网对智能家居的技术贡献

RFID 和 EPC	RFID 和 EPC 技术是物联网中让物品“开口说话”的关键技术，物联网中，通过 EPC，RFID 标签上存储着规范而具有互用性的信息，自动采集到中央信息系统，实现物品（商品）的识别
传感控制技术	传感控制技术是关于从自然信源获取信息，并对之进行处理、变换和识别的一门多学科交叉的现代科学与工程技术，它涉及传感器、信息处理和识别的规划设计、开发、制造、测试、应用及评价改进等活动
无线网络技术	物联网中，物品与人的无障碍交流，必然离不开高速、可进行大批量数据传输的无线网络；无线网络既包括允许用户建立远距离无线连接的全球语音和数据网络，也包括近距离的蓝牙技术和红外技术
组网技术	组网技术就是网络组建技术，分为以太网组网技术和 ATM 局域网组网技术，也可分为有线、无线组网，在物联网中，组网技术起到“桥梁”的作用
人工智能技术	人工智能是研究使计算机来模拟人的某些思维过程和智能行为（如学习、推理、思考、规划等）的技术；在物联网中，人工智能技术主要负责将物品“说话”的内容进行分析，从而实现计算机自动处理

本书将智能家居内容拆分成各个模块，力求在模块中体现布线、传输、控制、智能等方面的基础知识，使学生能够编写自己的数字家居小程序。

2. 智能家居的起源、定义与发展

1984 年，美国联合技术建筑系统（United Technologies Building System）公司将建筑设备

信息化、整合化概念应用于美国康涅狄格（Conneticut）州哈特福德（Hartford）市的“都市办公大楼”（City Place Building），通过对该座旧式大楼进行一定程度的改造，再采用计算机系统对大楼的空调器、电梯、照明等设备进行监测和控制，并提供语音通信、电子邮件和情报资料等方面的信息服务，这就诞生了世界上首栋“智能家居”建筑。此后，加拿大、欧洲、澳大利亚和东南亚等经济比较发达的国家先后提出了各种智能家居的方案，其中，德国弗劳恩霍研究会建成世界上第一座智能家居样板房，向人们揭示了未来住宅的前景和计算机技术新的发展趋势。最著名的智能家居要算比尔·盖茨的豪宅。比尔·盖茨在他的“未来之路”一书中以很大篇幅描绘他正在华盛顿湖建造的私人豪宅。他描绘的住宅是“由硅片和软件建成的”并且要“采纳不断变化的尖端技术”。经过 7 年的建设，1997 年，比尔·盖茨的豪宅终于建成。他的这个豪宅完全按照智能住宅的概念建造，不仅具备高速上网的专线，所有的门窗、灯具、电器都能够通过计算机控制，而且有一个高性能的服务器作为管理整个系统的后台。

智能家居也叫数字家庭，或称智能住宅，在英文中常称为 Smart Home，在我国香港、台湾等地区还有数码家庭、数码家居等叫法。通俗地说，智能家居是利用先进的计算机、嵌入式系统和网络通信技术，将家庭中的各种设备（如照明系统、环境控制、安防系统、网络家电）通过家庭网络连接到一起。一方面，智能家居将让用户有更方便的手段来管理家庭设备，比如，通过无线遥控器、电话、互联网或者语音识别方式控制家用设备，更可以执行场景操作，使多个设备形成联动；另一方面，智能家居内的各种设备相互间可以通信，不需要用户指挥也能根据不同的状态互动运行，从而给用户带来最大程度的高效、便利、舒适与安全。此外，智能家居还是以住宅为平台，兼备建筑、网络通信、信息家电、设备自动化，集系统、结构、服务、管理为一体的高效、舒适、安全、便利、环保的居住环境。

因此，智能家居可以定义为一个过程或者一个系统。利用先进的计算机技术、网络通信技术、综合布线技术，将与家居生活有关的各种子系统有机地结合在一起，通过统筹管理，让家居生活更加舒适、安全、有效。与普通家居相比，智能家居不仅具有传统的居住功能，提供舒适、安全、高品位且宜人的家庭生活空间；还由原来的被动静止结构转变为具有能动智慧的工具，提供全方位的信息交换功能，帮助家庭与外部保持信息交流畅通，优化人们的生活方式，帮助人们有效安排时间，增强家居生活的安全性，甚至为各种能源费用节约资金。

总之，在国家宏观发展需求（即建设节能型社会和创新型社会的目标）、信息技术应用需求（即信息化已成为当今人们生活的重要部分）、公共安全保障需求（即安全保证是衡量社区住宅环境的标准）和建筑品牌提升需求（即智能化是现代建筑灵魂核心的充分体现）以及其他主、客观因素的作用下，智能家居的产生是必然的。

随着家居控制技术的逐渐成熟，智能家居越来越普及。由于不同国家的不同国情，全球智能家居的风格也有多样化的特点。

美国智能家居偏重于豪华感，追求舒适和享受；德国的智能家居则继承了德国的一贯思想注重基本的功能性，追求专项功能的开发与应用；在澳大利亚，智能家居控制系统的特点是让房屋做到百分之百的自动化，而且不会看到任何手动的开关，安全问题也是考验智能家居的标准之一，澳大利亚智能家居保安系统里的传感器数量更多，即使飞过一只小虫，系统都可以探测出来；日本的智能家居以人为本，注重功能，兼顾未来发展与环境保护；韩国智

能家居采取实用主义，政府对智能小区和智能家居采取多项政策扶持，规定在首尔等大城市的新建小区必须具有智能家居系统，目前韩国全国80%以上的新建项目采用智能家居系统。

国内的智能化住宅和智能化家居虽然起步比较晚，但发展速度很快。目前国内几种智能家居系统有海信公司的DNet-home数字家庭、清华同方公司的e-Home数字家园、海尔公司U-home、西南交通大学科技公司开发的NDT系统、科龙公司的“现代家居信息服务集散控制系统”、美的公司智能家居系统、慧居公司全数字化网络智能家居终端、聚晖公司的智能生活专家——KOTI和北京科技大学的物联网智能家居系统等。

1.3 智能家居的市场需求和意义

1. 普通用户需求分析

现代人工作生活压力大，工作节奏快。上班还牵挂着家中老人、小孩是否安好，还在想家中的煤气有没有关牢。智能家居控制系统能够让用户轻松面对工作、生活，让人找到家的感觉，体现了家的和美与温馨。

当人们回到家中，温度、音乐、灯光、影院、屏幕…所有元素仿佛都有人操作，呈现出绝伦的和谐感。坐在舒适的沙发上，观赏倾心的电影，或聆听钟爱的乐曲，壁灯的光照分寸恰到好处，窗外夜色澄清。

智能化带给人们的是前所未有的自由。动动手指头，甚至什么都不用做，一切细节都在畅快自如地围绕在用户身边流淌。

2. 老龄化社会需求分析

随着传统的工业化社会向现代信息化社会演进，住宅作为一种物质和社会现象的人类居住生活空间，最新的、未来的科技都在其中展现着梦想的张力。在新的世纪里，当人们对未来生活的理解开始变成社会共识的时候，智能化、可持续发展将逐渐成为未来住宅的两个具体要素。

中国将逐渐进入老龄化社会，老年人数量连年增长，年轻人数量却远远不足，届时将出现数量可观的“空巢老人”，和更多的需要照顾的儿童。社会基础设施建设，不再是传统意义上的设施建设，而是要创新设备，使新时代的设备能够具有更广泛的互联互通性和更智能的服务性，一言以蔽之，未来社会建设需要我们的生活环境更加和谐，更加智能化，为了解放年轻劳动力、更好地照顾老年人而努力。

3. 行业的影响分析

智能家庭中的行业应用就是利用RFID、传感、定位、通信等技术，通过承载网络，包括互联网、移动通信网、广播电视网、传感网，实现物与物（T2T）、物与人（T2H）的连接，从而实现家庭对家电设备及应用的智能化管理、控制。

物联网大潮下的智能家居行业在我国乃至全世界都有广阔的前景，是一个朝阳的行业，就目前的发展趋势分析，预计在今后的几年内全世界将有近亿家庭构建智能、舒适的家居生活。

4. 行业发展分析

政府各部门态度积极，提供了很好的宏观环境。

我国高层政府的一系列讲话、报告、措施和相关政策明确表明，大力发展物联网产业将

是我国一项具有国家战略意义的重要决策，各级政府部门将会大力扶持物联网产业的发展，将物联网的建设发展上升到国家战略的地位。国家政府各部门的大力支持，将有利地驱动物联网产业的快速发展和进程。在物联网的应用中，智能家居产业的应用受众量最大，政府的明朗态度将大力推动智能家居物联网技术的应用。

（1）物联网核心技术发展迅速

物联网发展的核心力量是其技术的发展，依靠我国自身的核心技术将使其产业链逐步摆脱外界进口设备厂商带来的成本高、标准不统一的问题。包括 RFID、ZigBee、云计算等在内的物联网发展的关键技术在今后都会有进一步的突破和应用，相关标准的制定也会使技术的统一和推广更加完善。这将从根本上扩大物联网技术在智能家居产业的应用和发展。

（2）大量资本被吸引投入

国家政策对物联网的扶持吸引了更多新资本的涌入，更多的企业开始向物联网领域投入资金，寻求新的市场空间和发展机遇。

（3）家庭宽带的普及将促进智能家居的使用

在物联网时代，我们将遇到大量的数据量，家庭宽带接入是智能家居应用于家庭智能化的前提，随着家庭宽带的应用越来越广泛，将会促进智能家居的广泛应用。

（4）智能家居意义分析

与普通家居相比，智能家居不仅具有传统的居住功能，提供舒适安全、高品位且宜人的家庭生活空间；还由原来的被动静止结构转变为具有能动智慧的工具，提供全方位的信息交换功能，帮助家庭与外部保持信息交流畅通，优化人们的生活方式，帮助人们有效安排时间，增强家居生活的安全性，甚至为各种能源费用节约资金。

（5）智能家居从梦想走入现实

智能家居产品为家居环境、家电设备提供一个共享的接入中心，实现对家庭环境及其设备的智能管理、远程管理、集中管理和资源共享。随着网络科技的高速发展，在可以预见的未来，在智能化住宅中，以宽带网络将家里的电脑、电视机、家电、安防系统等连成一体的自主控制、扩展、享用的工作、学习、娱乐家庭综合信息服务平台。

（6）智能家居绿色节能

全世界能源消耗总量中有 40% 是建筑的能源消耗，其中住宅能源消耗又占了 2/3。随着石油、煤炭等能源的日益枯竭，人们对能源现实的理解也日益加深，融入自然、节能、可持续发展的住宅设想也越来越成为未来住宅在智能化之外的另一个重要指标。未来的智能化住宅则为人们营造了一个舒适、安全、方便和高效率的生活环境。

（7）智能家居有益健康

健康问题是年轻人的潜在问题，是老年人的首要问题。针对未来人们的养老、就医压力，智能家居能够检测常规健康问题，减少潜在疾病危险，同时还能为养老助老提供良好的空间，是未来社会智能化社区的一个重要方面，是未来国家为之努力的方向之一。

1.4　智能家居实验室系统设计

1. 总体描述

实训实验的智能家居实验室，表现为利用信息传感设备（同居住环境中的各种物品松

耦合或紧耦合）将家居生活有关的各种子系统有机地结合在一起，并与互联网连接起来，进行监控、管理信息交换和通信，实现家居智能化，包括：智能家居控制管理系统、终端（家居传感器终端、控制器）、家庭网络、信息中心等。

物联网智能家居要将自成体系、互不相连的子系统协调起来，就必须有一个兼容性强的中央家居处理平台，接收并处理控制设施发出的信息，然后传送信号给希望控制的家电或者其他家居子系统。

可以将中央处理平台形象地理解为一个交通警察，它的职能就是在家庭智能局域网中，引导和规划家居子系统中的各种信号，有了它，可以通过键盘、触摸屏、电话，或者手持无线遥控设备来和家居子系统进行快速的沟通。中央家居处理平台还必须具有良好的扩展性能，以满足用户在使用过程中不断增长的需求点。

智能家居系统将家庭中各种与信息相关的通信设备、家用电器和家庭保安装置，通过家庭总线技术连接到一个家庭智能化系统上进行集中的或异地的监视、控制和家庭事务性管理，并保持这些家庭设施与住宅环境的和谐与协调。这些功能都是通过智能家居系统中的家庭网络控制器来实现的，家庭网络控制器具有家庭总线系统，通过家庭总线系统提供各种服务功能，并和住宅以外的外部世界相通连。可以这样说，智能家居系统是智能住宅的核心。由此可见，智能家居系统在智能住宅中的重要地位。

2. 设计原则

无规矩不成方圆。智能家居的基本设计原则是分层式设计。把智能家居系统分为感知控制、网络传输和智能控制几个层次。每个层次之间相互独立，又为相邻层提供服务。分层式设计原则如图 1-1 所示。

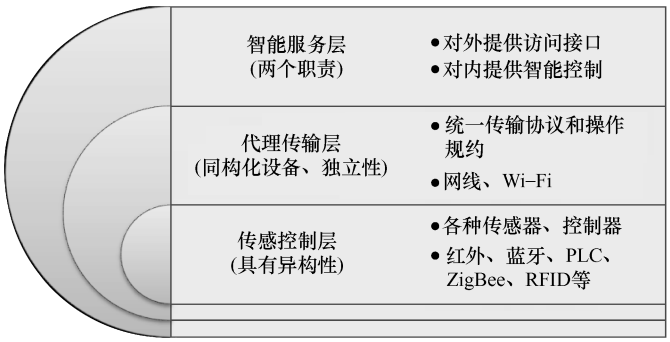


图 1-1　智能家居分层式设计原则

3. 系统概览与简介

本智能家居含有以下八个子系统：

1）综合布线系统；

2）家电控制系统；

3）环境信息采集系统；

4）远程抄表系统；

5）安防监控系统；

6）人机交互系统；

7）健康关护系统；

8）智能服务系统；

系统之间的关联结构如图 1-2 所示。

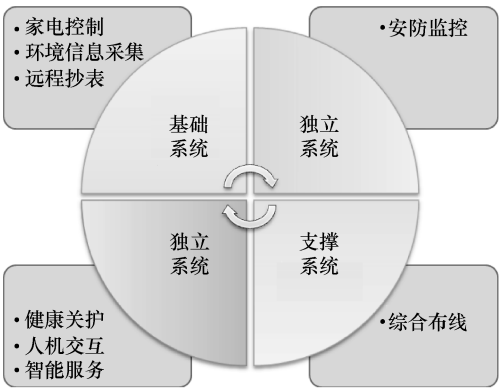


图 1-2　智能家居系统关联结构

其中综合布线系统为底层通信介质的基础，直接为家电控制、环境采集、远程抄表系统、安防监控系统提供硬件连接通道，直接或间接地为人机交互系统、健康关护系统、智能服务系统提供底层通信基础。

家电控制系统、环境采集系统、远程抄表系统、安防监控系统是数字家庭基础的功能设施，是智能家居必须实现的内容。

人机交互系统、健康关护系统自成体系，它们既需要其他系统的支持，也向高级服务提供依据，同时也可以独立运行。

智能服务系统是整个智能空间的高级实现形式，是整个智能家居系统的核心和最终目的。它依赖其他系统，又高出其他系统，它无法脱离其他系统存在，但它面对的却是服务流和服务数据，并不直接针对特定设备。

这几个系统的实物连接框图如图 1-3 所示。

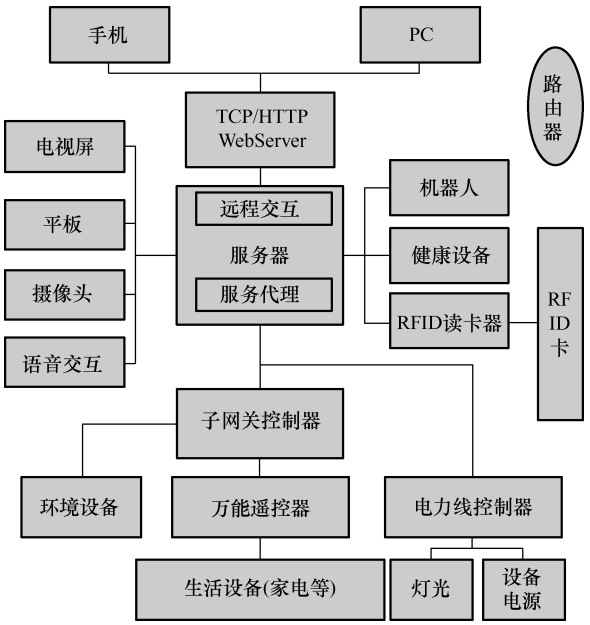


图 1-3　智能家居实物连接框图

本系统是一个层次化的系统，层次图如图 1-4 所示。

一个好的系统设计是松耦合的、模块化的、标准化的。本系统为了让不同的设备开发者各司其职，同时又能在一定程度上互相沟通，设计了层次化的系统结构。

1）异构设备层：家庭设备千变万化、通信手段多种多样，因此家庭设备具有异构的特点。承认设备的异构化，能够极大扩展设备的选型，方便硬件开发者开发新设备。因此，系统设计的最底层规定为异构设备层，给开发者提供相当程度的自由。

2）硬件接入与 Wi-Fi 硬件层：异构设备层为我们带来了大量优秀的硬件，但是这些硬件是孤立的，和服务系统的衔接并不友好。于是系统扩展了异构设备层，把它们发展成了 Wi-Fi 硬件层。顾名思义，系统通过提供 TTL/RS232 转 Wi-Fi 模块，规定所有的设备信号，最终都必须通过 Wi-Fi接入到路由器，所有上行信号和控制信号都重定向到 Wi-Fi 输出。TTL/RS232 转 Wi-Fi 模块是一个透明传输的硬件模块，它能够在硬件层直接接入到 TTL/RS232 接口，同时可以自动连接路由器，并能够屏蔽硬件差异，在上述两种不同的硬件接口之间透明传

<table>
<tr><td colspan="4">服务表现层</td></tr>
<tr><td colspan="2">HTTP服务层</td><td>Socket服务层</td><td>WebService服务层</td></tr>
<tr><td colspan="4">服务管理层</td></tr>
<tr><td>服务标准化模块</td><td>服务标准化模块</td><td>服务标准化模块</td><td>服务标准化模块</td></tr>
<tr><td colspan="4">Wi-Fi布线层</td></tr>
<tr><td colspan="2">硬件接入层</td><td colspan="2" rowspan="2">Wi-Fi硬件层</td></tr>
<tr><td colspan="2">异构设备层</td></tr>
</table>

图 1-4　智能家居逻辑层次

输、转发数据，因此硬件开发者不需要额外制作硬件或编程序，就能把设备无缝转换成 Wi-Fi 家居设备。

3）Wi-Fi 布线层：顾名思义，即把所有的设备通过 RJ45 网线或是 Wi-Fi 连接起来，使用 TCP/UDP 对话。

4）服务标准化模块：前几个模块只保证了设备能够通过 Socket 访问，并不能保证设备信息是否被正确理解。这就需要对设备信息进行编码和解码。要求硬件信息强制符合某种编码规范，是不切实际的，这会极大地造成硬件开发的困难。一个变通的方法是，在软件层，对不同设备传输的信息进行理解，进行编解码，每一种设备对应一个服务标准化模块，每个服务化标准模块都和服务器之间遵循统一的通信协议，以达到接收异构信息，同时“归一化”设备信息的目的。

5）服务管理层：服务管理层是系统的核心。经过服务标准化模块，服务管理层“看到的”是一个一个标准的设备（事实上是标准的服务模块），这样就可以统一地管理底层设备了。服务管理层负责设备管理、信息采集、数据存储、数据分析，智能服务等一系列服务功能，是整个智能家居的“大脑”和“记事本”。

6）Socket、HTTP、WebService 服务层，提供了不同的服务访问手段，针对不同终端服务访问设备工作在网络七层协议不同层的现状，提出的一种兼容并包的手段。

7）服务表现层：服务表现层主要责任是实现丰富的用户交互方式，例如手机界面、网页访问、语音对话、行为理解、机器人管家等。

一个更加抽象的系统层次图如图 1-5 所示。

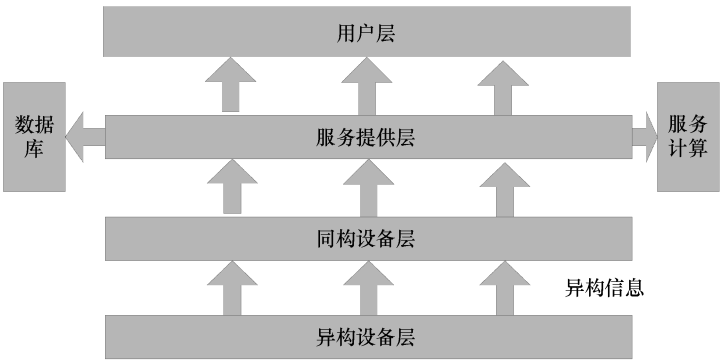


图 1-5　更加抽象的智能家居逻辑层次

1.5　智能家居系统功能

1. 便捷的生活服务

（1）智能无线遥控系统

自学习式智能遥控器，实现对所有电器、灯光、窗帘和部分环境设备的遥控，以及一键式场景控制。

（2）环境监测与采集系统

环境监测与家电控制子网关。随时获取和上传家庭的光照、温度、湿度、噪声、烟雾、有害气体等信息；同时还能代为控制智能遥控器。

（3）抄表系统

使用 RS485/Wi-Fi 实现电表等的读取。

（4）电力线系统

继电器与电力线载波控制器。能够通过电力线传播控制信号，免去了布线之苦。

2. 安心的环境检测

（1）环境感知

有害气体、烟雾、煤气传感器，通过子网关控制器采集。

（2）安防门禁

1）RFID 卡系统，确认来客身份。

2）网络摄像机，提供局域网视频访问和广域网图片访问。

3）文件服务器，用于存储视频监控录像。

3. 和谐的人机交互

（1）大屏幕交互

智能电视和投影仪。可以接受服务器调用，及时向住户推送消息。

（2）语音交互

1）通过手机或送话器，向服务中心请求简单的命令。

2）音响系统能够以语音方式告知控制结果，或是主动提醒用户。

（3）机器人交互

1）机器人具有自主移动功能，内置摄像头，能够监控移动视频信息。

2）机器人能够和住户进行语音交互。

3）机器人能够控制家电。

4. 体贴的健康服务

（1）健康设备的开发与接入

设备能够采集心电图、血压、血糖等信息，并上传到服务器。

（2）健身监察设备

该设备可以记录用户一天的运动量，上传到服务器。

（3）健康信息提醒

将提醒信息推送到手机。

5. 自发的智能服务

以状态参量为基础的联动和服务计算。

（1）服务发布网关

1）将家庭服务发布到 Internet，供手机、PC 等访问。

2）提供语音、图像等高级服务。

3）为服务计算服务器组提供状态数据。

（2）服务计算服务器组

1）健康数据分析服务

2）用户喜好分析服务

1.6 智能家居系统设备实物

1. 网络设备

网络设备主要包含服务器、二层交换机、路由器等。服务器负责提供基于网络的服务和对家居设备的控制，交换机、路由器则构成了整个家居的基础网络。如图 1-6 ~ 图 1-8 所示。

图 1-6 服务器

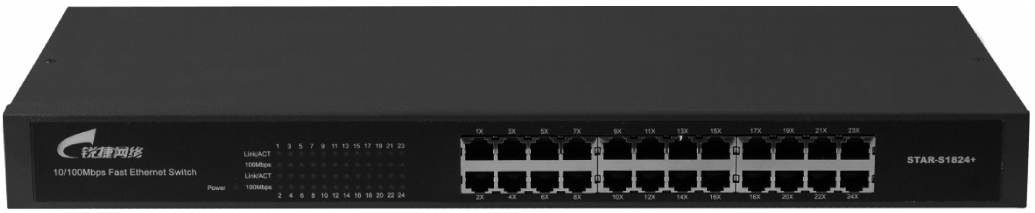


图 1-7 二层交换机

2. 家电控制与子网关设备

家电控制与网关设备是一个处于下层的控制中心，它们负责采集底层传感器的信息、控制底层执行器运作，同时提供一个基于 Wi-Fi 的 TCP 访问接口，使服务器能够通过无线路

由器和它们交互，完成高层的服务行为。具体设备如图 1-9 和 1-10 所示。

图 1-8　无线路由器

图 1-9　智能家庭信息子网关

3. 环境感知设备

环境感知设备即网络传感器设备，包括视频摄像头，温、湿度传感器，照度传感器，烟雾传感器，红外人体传感器等。其中摄像头提供了家居视频图像；温、湿度传感器，照度传感器负责采集环境信息，并转化为数字信号传送；烟雾传感器、红外人体传感器则会产生一个开关量信息，参与到家庭安防中去。具体设备如图 1-11 ~ 图 1-14 所示。

图 1-10　家电设备集成控制器（万能遥控器）

图 1-11　网络摄像头

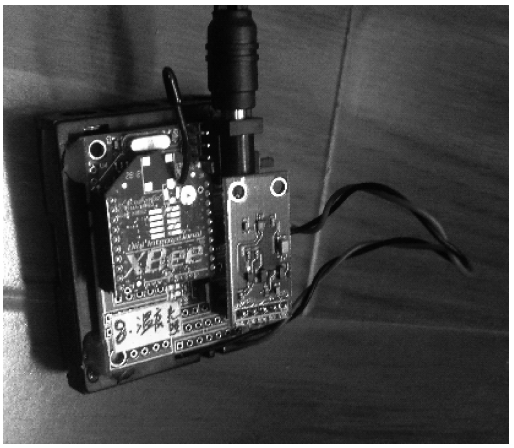


图 1-12　温、湿度，照度传感器（ZigBee 通信方式）

4. 家电与灯光设备

家电与灯光设备是智能家居必不可少的一部分，它们都是基于红外控制的设备。然而目

前市场上各个家电厂商之间并没有就统一的家电控制接口达成一致，甚至没有提供控制接口，给智能家居的家电接入造成了困难。为此本实训设计了家电设备集成控制器，这些控制器能模拟红外遥控器，为了实现零布线、不改动已有家电的优点，灯光系统连接在红外插座上，也是间接由红外信号控制的。具体设备的图片如图 1-15 ~ 图 1-17 所示。

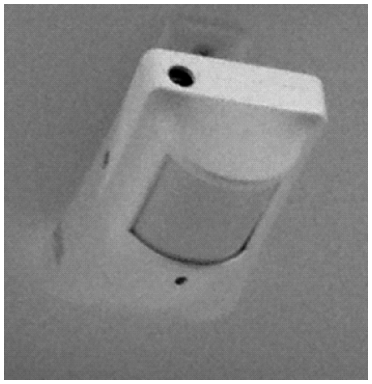


图 1-13　红外感应设备（门禁监测）

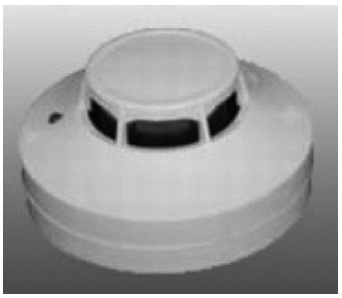


图 1-14　红外烟雾传感器

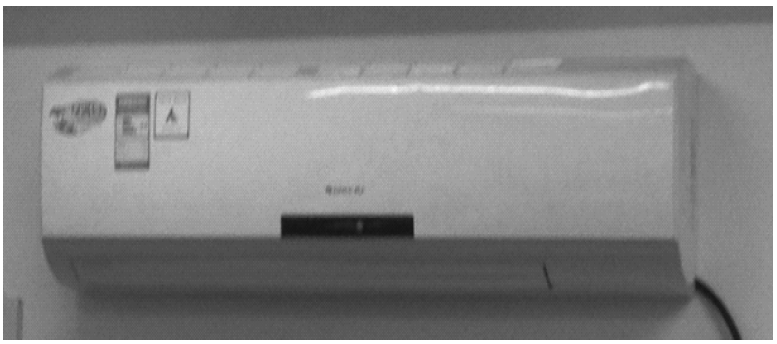


图 1-15　空调设备

图 1-16　创维电视

图 1-17　灯光设备

5. 健康设备

健康设备主要包括便携式人体健康采集设备，它们通过蓝牙或是 ZigBee 和健康网关、健康电视通信，上传健康数据到服务器。主要的设备如图 1-18 ~ 图 1-23 所示。

图 1-18　血压计

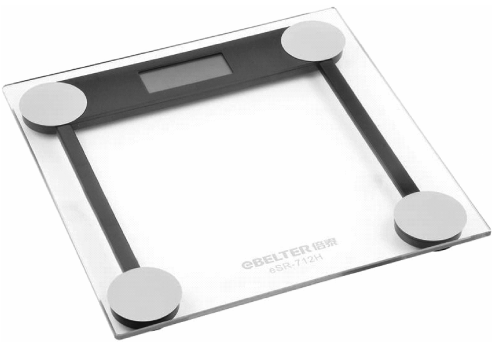


图 1-19　体重秤

图 1-20　脂肪测量仪

图 1-21　多功能床（可配合多种健康设备检测心率等）

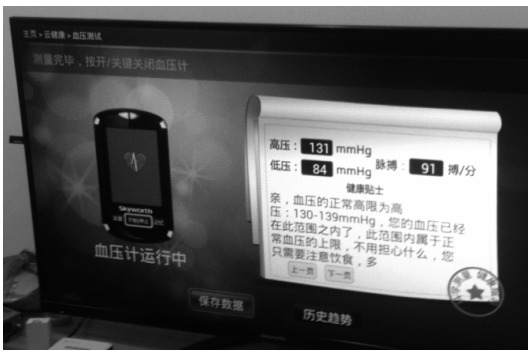


图 1-22　无线血压仪测量结果在云电视中显示

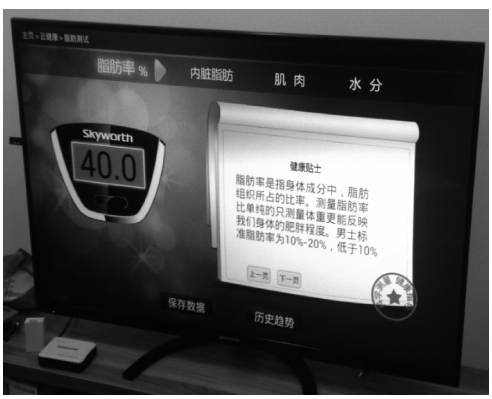


图 1-23　无线脂肪仪测量结果在云电视中显示

6. 网络终端设备

网络终端设备包括手机、平板电脑、机器人等，它们主要负责人机交互部分。主要的设备如图 1-24 ~ 图 1-27 所示。

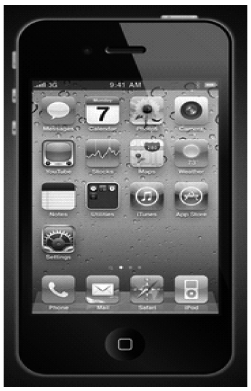


图 1-24　手机

图 1-25　安卓系统的平板电脑

图 1-26　Win8 系统的平板电脑

图 1-27　机器人

以上介绍了智能家居的各个设备，这些设备有机地连接在一起，各司其职又互相协作，才能形成一套完整的智能家居控制体系，设备之间的连接可以参照图 1-28。

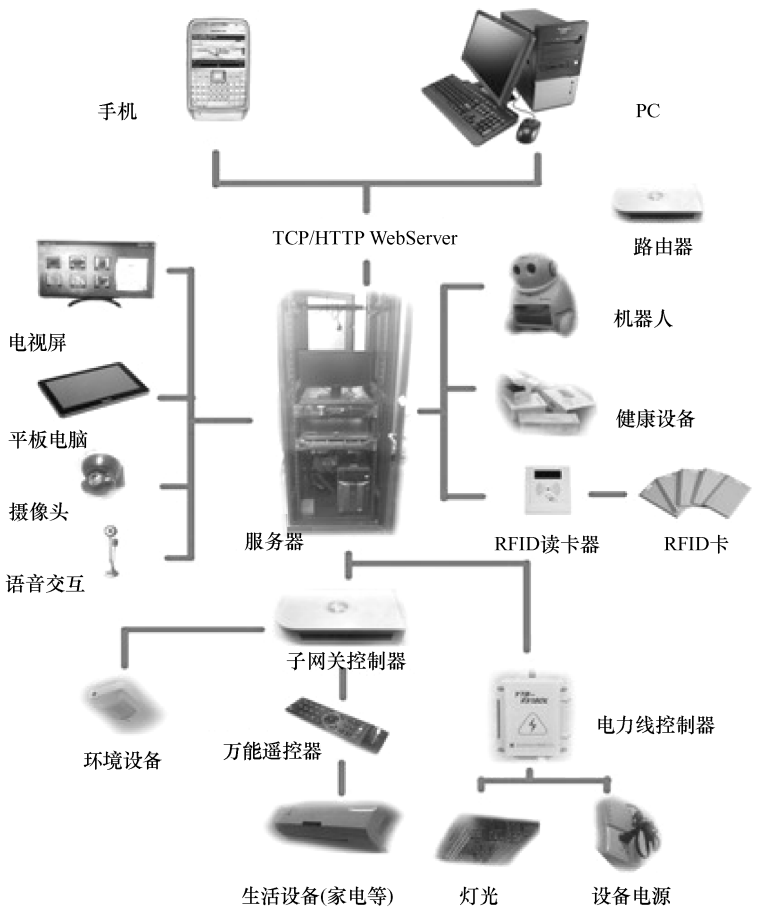


图 1-28　智能家居设备连接实物图

1.7　小结

本章讲述了一个完整的物联网智能家居系统的结构和功能，智能家居的技术几乎涉及了信息技术的方方面面，是一种聚合性、系统性的创新应用与发展，其本质主要体现在三个方面：一是网络特征，即需要智能家居的物体一定要能够实现互联互通的网络；二是识别与通信特征，即纳入智能家居的“物”一定要具备自动识别、物物通信的功能；三是智能化特征，即网络系统应具有自动化、自我反馈与智能控制的特点。

智能家居系统的构成包括功能构成和层次化构成：功能构成对应了智能家居环境中的各个具体功能模块，它是智能家居的基本构成条件；层次化构成则反映了智能家居系统间的相互联系和组织规则，它对智能家居的建设具有重要的指导意义。

智能家居的服务平台和各个子系统涵盖了从单片机、ARM 到移动操作系统、服务器的软硬件开发环境，从 C、C + + 到 Java、C#、ASP. NET 的软件编程语言，更有从 TCP/IP 到 WebService 的网络通信协议，甚至包括 LISP、Prolog 等产生式规则的推理系统，读者可以从各个角度出发，找到适合自己学习的内容，在以后的学习中，不断提高自己对物联网和智能

家居的理解以及相关技术水平，完成 CDIO 培养计划。希望本书能够帮助需要的人们学习物联网技术。

参 考 文 献

[1] 刘爱军. 物联网技术现状及应用前景展望［J］. 物联网技术，2012，01.
[2] 王志良. 物联网工程导论［M］. 西安：西安电子科技大学出版社，2011.
[3] 李莎. 物联网技术及智能家居应用研究［J］. 现代电子技术，2012，21.
[4] 张晓栋. 基于无线网络的家庭安防系统的硬件设计［D］. 武汉：华中科技大学，2008.
[5] 冯承金. 基于 ZigBee 和 ARM9 的智能家居系统的研究与设计［D］. 武汉：武汉理工大学，2010.
[6] 吕莉，罗杰. 智能家居及其发展趋势［J］. 计算机与现代化，2007，11.
[7] 覃朗. 基于 WindowsCE 的智能家居终端的设计与实现［D］. 重庆：重庆大学，2009.
[8] 沈星星. 基于 ZigBee 的智能家居系统关键技术的研究［D］. 南京：南京邮电大学，2011.
[9] 叶玮琼. 嵌入式 WEB Server 智能家居中央控制器实现方法研究［D］. 广州：广东工业大学，2005.
[10] 赵静. 嵌入式智能家居控制系统的研究与设计［D］. 武汉：武汉理工大学，2008.
[11] 范丽娜. 智能家居系统中家电控制的研究与实现［D］. 南京：南京邮电大学，2011.
[12] 马菁菁. ZigBee 无线通信技术在智能家居中的应用研究［D］. 武汉：武汉理工大学，2007.
[13] 沈舒海，王晓东，胡珊逢，等. 基于 GPRS 网络的智能家居控制系统［J］. 电子器件，2012，02.

第 2 章　家 庭 网 站

2.1　实验目的

1）熟悉 Microsoft SQL Server2008、Microsoft Visual Studio 2010。

2）了解如何从移动终端收发数据，并基于 .NET 网站进行数据的处理与展示。

3）了解 Socket 通信，数据库设计以及制作网站必备的 C#、HTML、CSS、JavaScript、JQuery、AJAX 等基础知识技能。

2.2　实验内容

搭建监护平台，设计基于安卓操作系统的手机客户端软件。通过监护平台对手持终端设备进行 GPS 定位，时时监控老人的位置信息，并对老人的行动数据进行分析，通过监护平台给老人发送提醒信息等。

2.3　实验所用仪表及设备

2.3.1　系统运行环境

1）硬件：安卓智能手机多部、Web 服务器一台、数据库服务器一台、数据采集服务器一台，对服务器的要求如下：

3.0（GHz）及以上处理器。

内存要求：8GB 或以上。

显卡要求：512MB 或以上显存的显卡。

硬盘要求：推荐 500GB 或以上。

网卡要求：千兆网卡。

2）软件：Microsoft SQL Server2008、Microsoft Visual Studio 2010、IIS。

2.3.2　开发技术

1）编程语言：C#；

2）地图服务：百度地图；

3）数据通信：TCP，Socket，AJAX；

4）网站制作：HTML、CSS、JavaScript（JQuery）；

5）数据库：Microsoft SQL Server 2008。

2.4 实验原理

我国人口老龄化加剧、空巢老人激增，该系统可满足子女、亲友对老人关爱服务方面的需求，让老人安享晚年。此部分服务主要由监护平台和手持终端设备组成。通过手持终端设备的GPS定位功能，可对老人的位置进行实时监控，遇到突发情况有自动报警功能。同时还具备很多个性化服务，如主动发送提示信息，终端设备自动进行语音播报；对重要的事件进行定时提醒，如按时吃药、好友生日提醒等；设置个性化铃声以及背景，让老人不会感到孤独。

具体功能如下：

1）GPS定位，对老人的位置进行实时监测。

2）对老人行动数据的分析，包括日行距离以及能量消耗等。

3）个性化的提醒服务，通过监护平台可给老人发送提示消息。

4）提供历史数据的查询。

5）老人活动轨迹回放。

6）健康档案的建立及存储。

7）大的按键和字体显示，方便老人使用。

8）社交圈信息的保存及调用、分析和处理。

9）提供用于管理员和独立用户分别访问的网站权限。

10）提供多样化的浏览功能（如同时显示或报警优先、轮循监控）。

老人监护系统整体框架如图2-1所示。

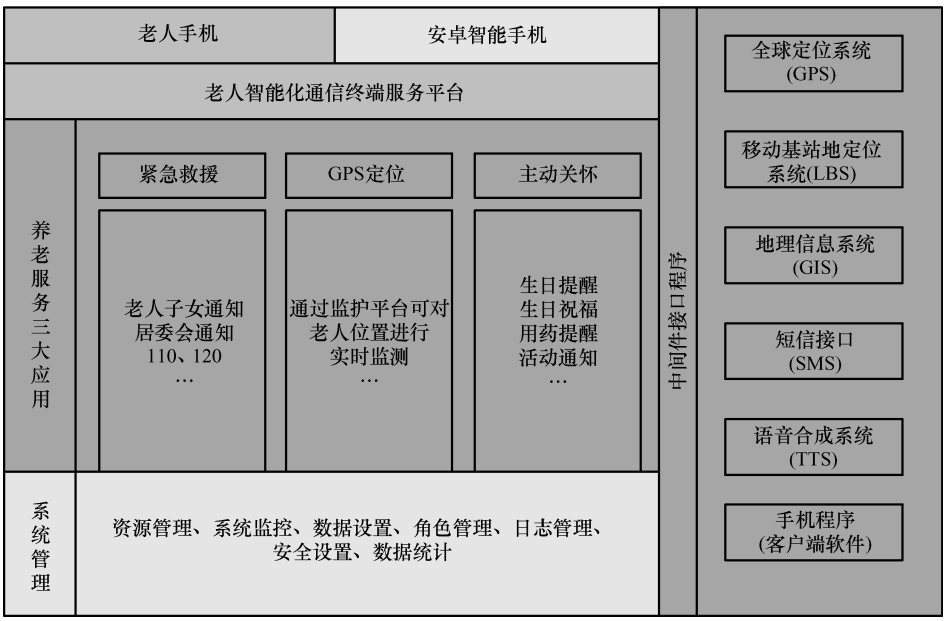


图2-1 老人监护系统整体框架

系统运行原理如图2-2所示。

老人随身携带有GPS功能模块的移动监控终端（如手机），老人的位置及状态信息将会

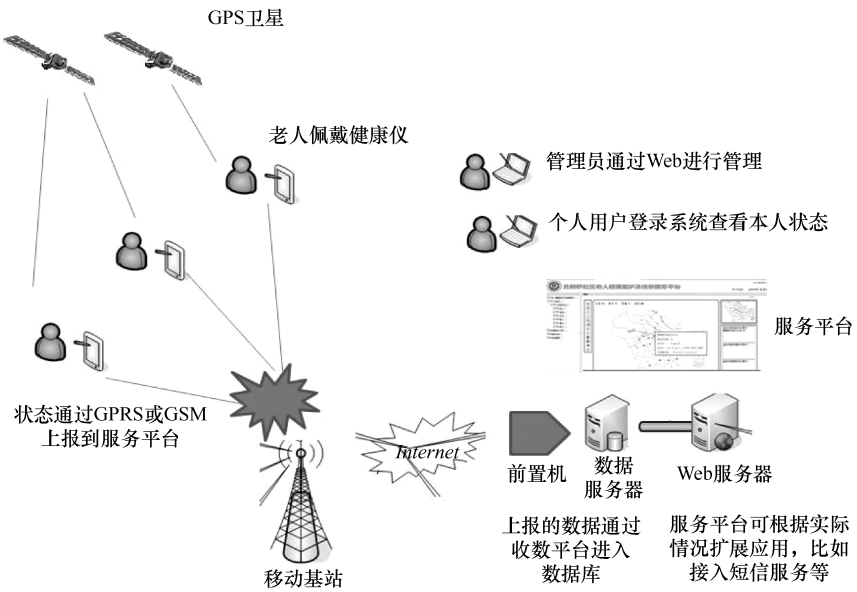


图 2-2　系统运行原理图

通过 GPRS/GSM 上报至社区监控服务平台，上报信息通过收数平台写入数据库，当管理员或用户访问网站时，前台页面将从数据库读取最新上传的数据，在 GIS 地图上标定出老人的状态信息。当监测到老人生理数据出现异常等情况时，系统给出自动报警，让老人得到紧急救护。系统中存储的老人健康档案及历史监测数据将给医护人员为老人的诊断与救治提供决策辅助。

2.5　实验程序设计

2.5.1　代码实现

1. 数据通信模块

Socket 通信模型如图 2-3 所示。

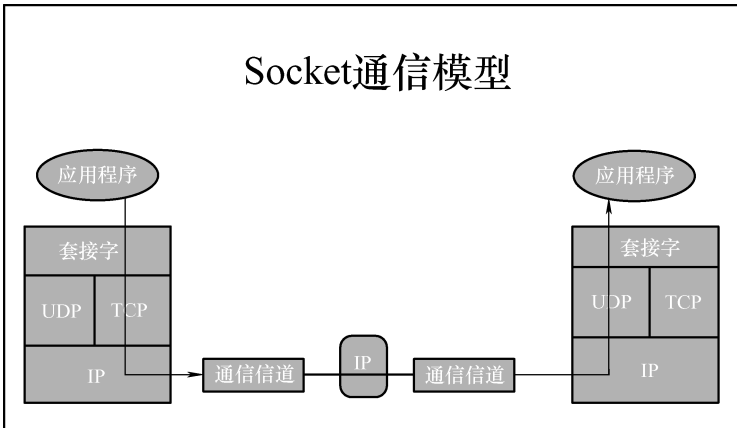


图 2-3　Socket 通信模型

```
//开始监听
IPAddress ip = IPAddress. Parse("202. 204. 54. 166");
TcpListener listener = new TcpListener(ip,8903);
listener. Start();
//传入连接
TcpClient client = listener. AcceptTcpClient();
//接收数据
private NetworkStream streamToClinet;
private const int BufferSize = 8192;
private byte[] buffer;
streamToClinet = client. GetStream();
buffer = new byte[BufferSize];
//异步
AsyncCallback callBack = new AsyncCallback(ReadComplete);
streamToClinet. BeginRead(buffer,0,BufferSize,callBack,this);
//回调
private void ReadComplete(IAsyncResult ar){. . . . . }
```

2. 地图坐标展现模块

```
var gpsPoint;
var bm = new BMap. Map("allmap");
bm. addControl(new BMap. NavigationControl());
bm. centerAndZoom("北京",15)
BMap. Convertor. translate(gpsPoint,0,translateCallback = function (point) {
        var marker = new BMap. Marker(point);
        bm. addOverlay(marker);
        var label = new BMap. Label("这是我的位置",{offset: new BMap. Size(20,
-10)});
        marker. setLabel(label);
        bm. setCenter(point);
}
```

3. 数据库操作

```
public static void UpdateGps(string id,string lng,string lat)
    {
        string cs = "server = . \\SQLEXPRESS;database = Gpsorlbs;uid = sa;pwd = sql";
        SqlConnection cnn = new SqlConnection(cs);
        cnn. Open();
        SqlCommand sqlc = new SqlCommand("UPDATE detail SET LNG ='" + lng +"',
LAT ='" + lat +"'WHERE ID ='" + id + "'",cnn);
        sqlc. ExecuteNonQuery();
```

```
            cnn. Close( ) ;
        }
```

4. AJAX

```
$ . ajax(
            {
              async: false,
              type: "post",
              url: "op/Request. aspx",
              data: {"id":we},
              success: function (data) {
                var index = data. indexOf(",") ;
                var xx = data. substring(0,index  -1) ;
                var yy = data. substring( index +1) ;
                gpsPoint = new BMap. Point( xx,yy) ;
              }
            }
    ) ;
```

2. 5. 2　界面效果

用户的信息管理，需要账号可登录平台，如图 2-4 所示。

图 2-4　登录页面

GPS 定位，将老人的位置在 GIS 地图上标定（见图 2-5），便于对其进行时时监测。
给老人发送亲情短信（见图 2-6），终端可语音播报。
支持历史数据的查看，如图 2-7 所示。
对老人行动数据进行分析（见图 2-8），可提供每天运动情况及能量消耗情况。

图 2-5 GIS 定位

图 2-6 短信提醒

2.5.3 数据库设计

数据库存储内容分以下五部分：

1）用户注册信息：包括用户名、密码、用户权限级别及编号。

2）老人健康档案。包括以下内容：

① 人口学信息：如姓名、性别、出生日期、民族、身份证件、文化程度、婚姻状况等；

② 社会经济学信息：如户籍性质、联系地址、联系方式、职业类别、工作单位等；

③ 亲属信息：如家庭成员姓名、是否跟老人同住等；

④ 社会保障信息：如医疗保险类别；

⑤ 基本健康信息：如血型、过敏史、预防接种史、既往疾病史、家族遗传病史、健康危险因素、残疾情况、亲属健康情况等。

⑥ 生活方式：如运动、饮食习惯，爱好等。

3）实时监测信息包括：

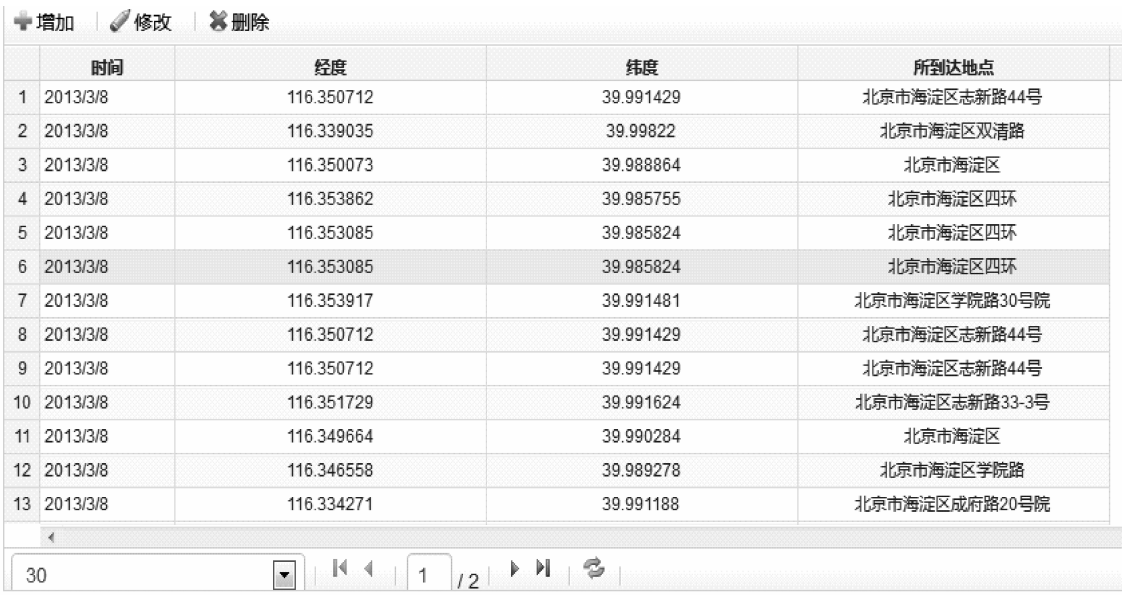


图 2-7　历史数据查看

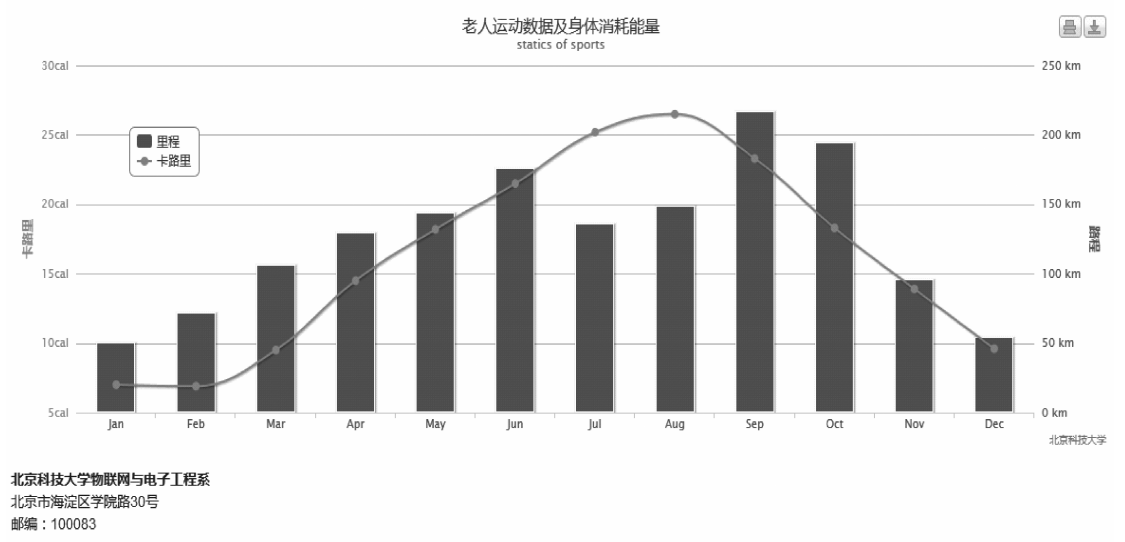


图 2-8　运动数据分析

① 运动相关信息：位置（经纬度）、移动速度、移动距离、消耗能量等。

② 生理参数：血压、血糖、脉搏、体温、呼吸等。

4）历史数据记录。

5）个性化信息提醒。

2.6 实验步骤

2.6.1 环境搭建

1. 安装 Microsoft SQL Server 2008 R2
2. 安装 Microsoft Visual Studio 2010
3. 启动 IIS 服务

1）打开控制面板，如图 2-9 所示。

图 2-9 控制面板

2）如图 2-10 所示，选一个程序和功能选项打开它。

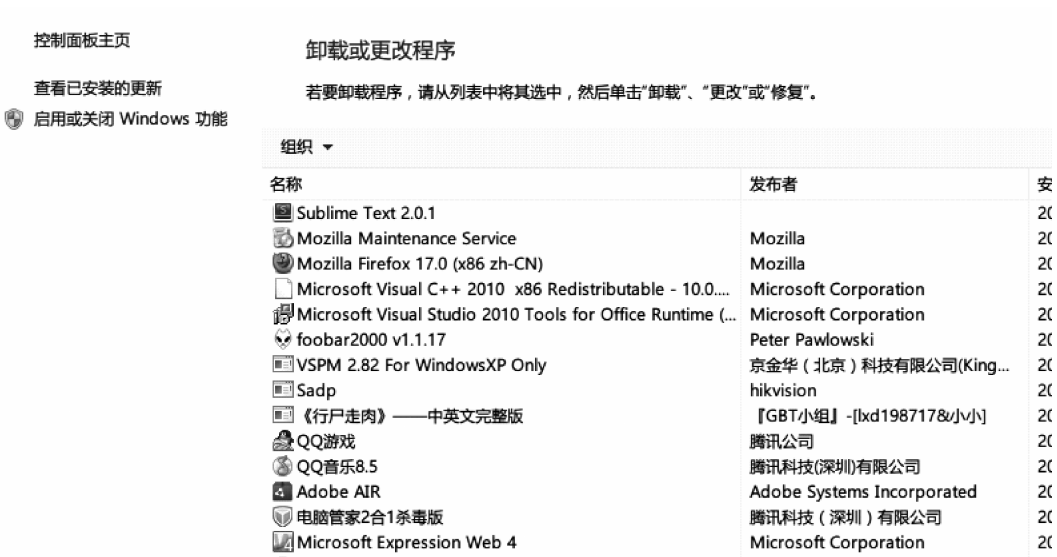


图 2-10 卸载或更改程序界面

3）如图 2-11 所示，启动或关闭功能打开，显示如图 2-11 所示对话框。

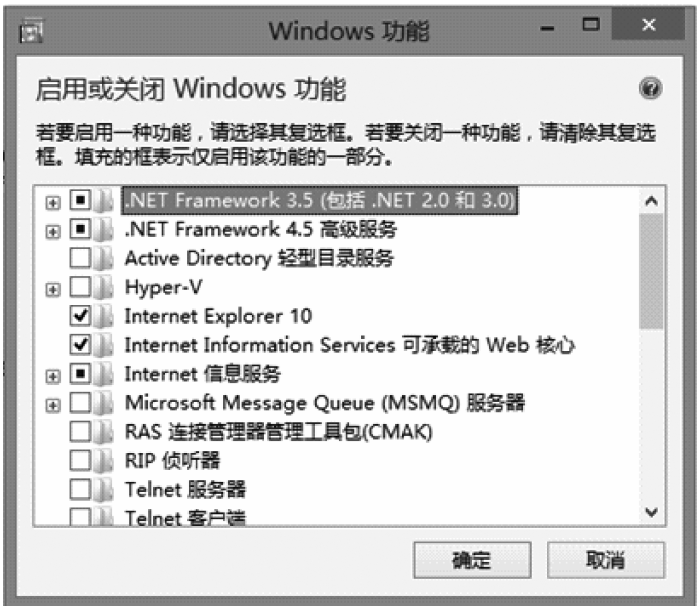


图 2-11　启动或关闭 Windows 功能界面

4）展开 Internet 信息服务，按图 2-12 所示选择，单击确定。

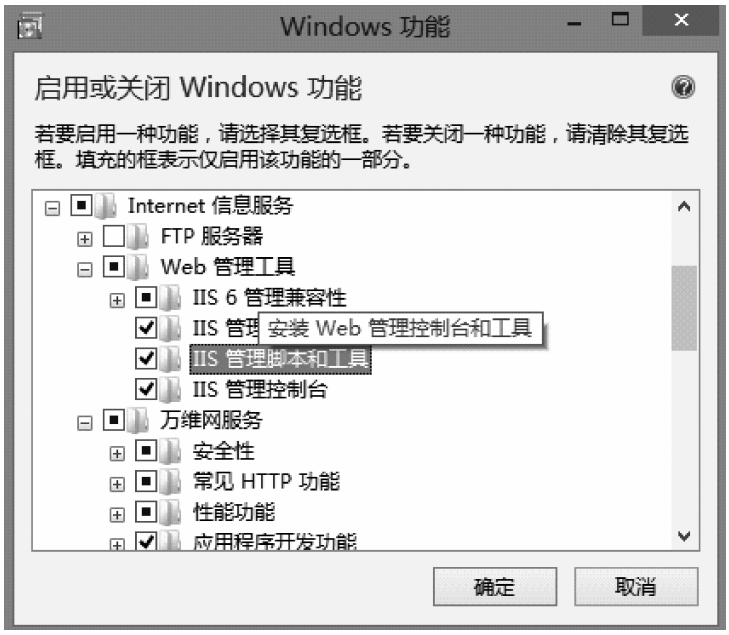


图 2-12　选择启动的 Internet 信息服务项

5）检测是否安装好 IIS。在自己的浏览器输入 127. 0. 0. 1，显示如图 2-13 所示信息即为成功。(图 2-13 为 Win7 系统，Win8 系统会显示 IIS8 的界面)。

这样我们通过 IIS 就可以发布一个网站了。

图 2-13 成功安装 IIS 的标志界面

2.6.2 在 SQL Server Management Studio 里建立数据库

1）打开 SQL Server Management Studio，在左侧的“对象资源管理器”里的“数据库”选项上单击鼠标右键，新建数据库，在弹出的对话框中设置数据库名及存储路径如图 2-14 所示。

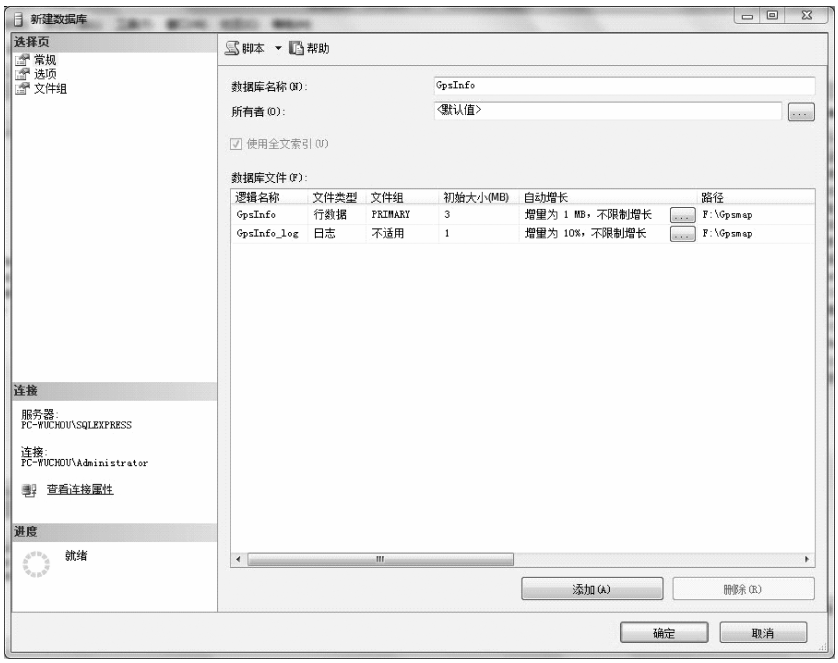


图 2-14 新建数据库

2）在新建的数据库的表选项上单击鼠标右键，新建表，设置表的字段名、数据类型及相关属性，如图 2-15 所示。

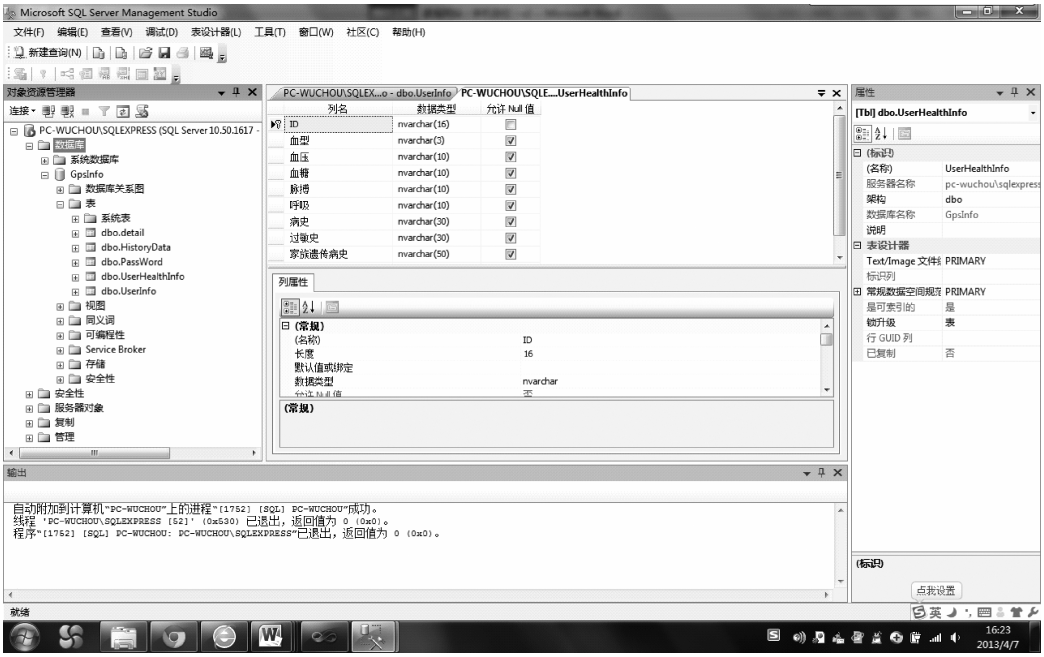


图 2-15　数据库存储内容

2.6.3　在 Visual Studio 2010 里创建数据通信窗体应用程序

1）首先打开 Visual Studio 2010，在首页选择文件—新建项目—Windows 窗体应用程序，新建窗体应用程序如图 2-16 所示。

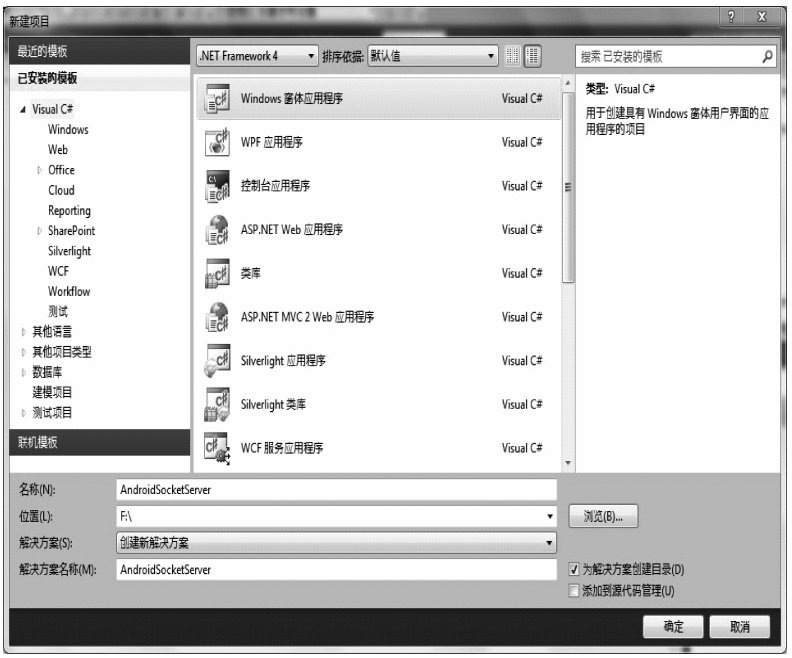


图 2-16　新建窗体应用程序

2）编写 C#程序代码。

DatabaseOp. cs 实现对数据库里老人位置信息的更新。

Global. cs 发送消息类的定义。

Process. cs 定义根据 Web 或安卓手机的不同客户端进行数据处理及解析 GPS 数据、计算两点之间距离（如用于计算老人离家距离）。

Program. cs 程序入口，建立通信连接。

RemoteClient. cs 定义客户端。

Select. cs 实现与 Web 或安卓手机的连接、通信。

窗体应用程序内容如图 2-17 所示。

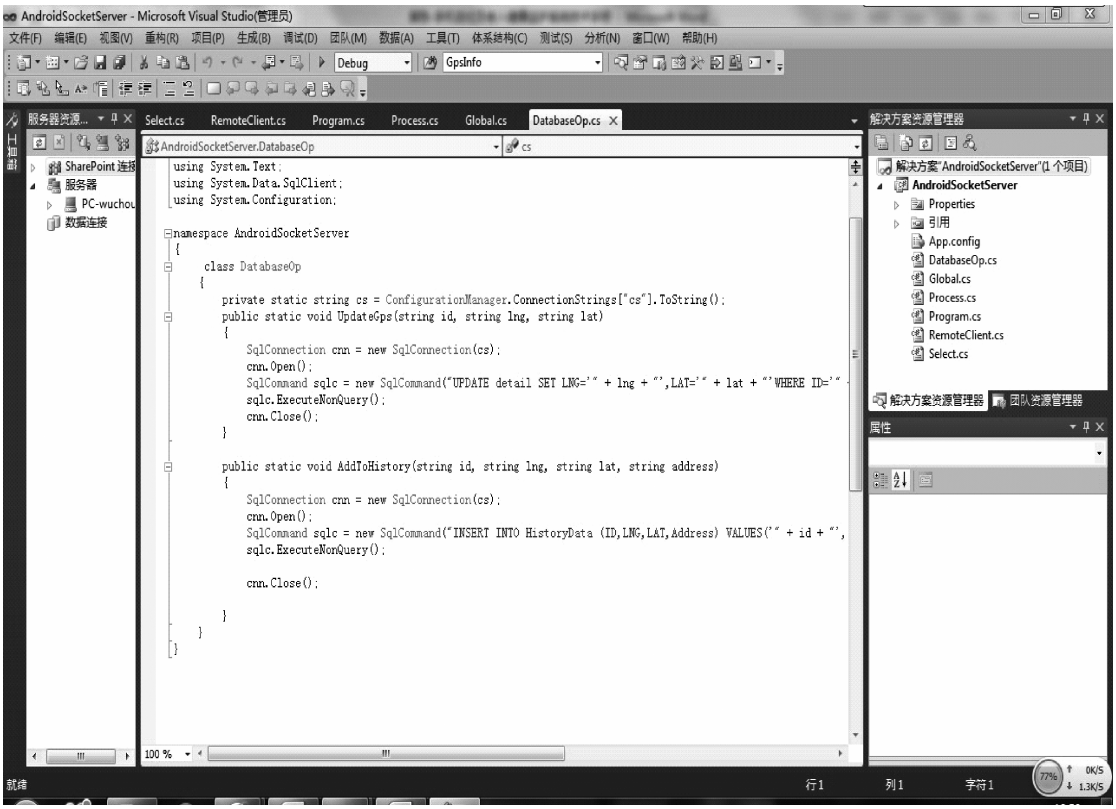


图 2-17　窗体应用程序内容

2. 6. 4　在 Visual Studio 2010 里创建网站

1）首先打开 Visual Studio 2010，在首页选择文件—新建—网站，如图 2-18 所示。

2）在弹出的对话框中选择新建 ASP. NET 空网站，如图 2-19 所示，这个可以根据需求不同而选择其他选项。

3）这样就创建了一个简单的网站工程，此时由于没有添加其他内容，所以只有显示一个配置文件，如图 2-20 所示。

4）编写前台界面、脚本程序，以及后台 C#程序，扩充网站内容。其中，图片、js 脚

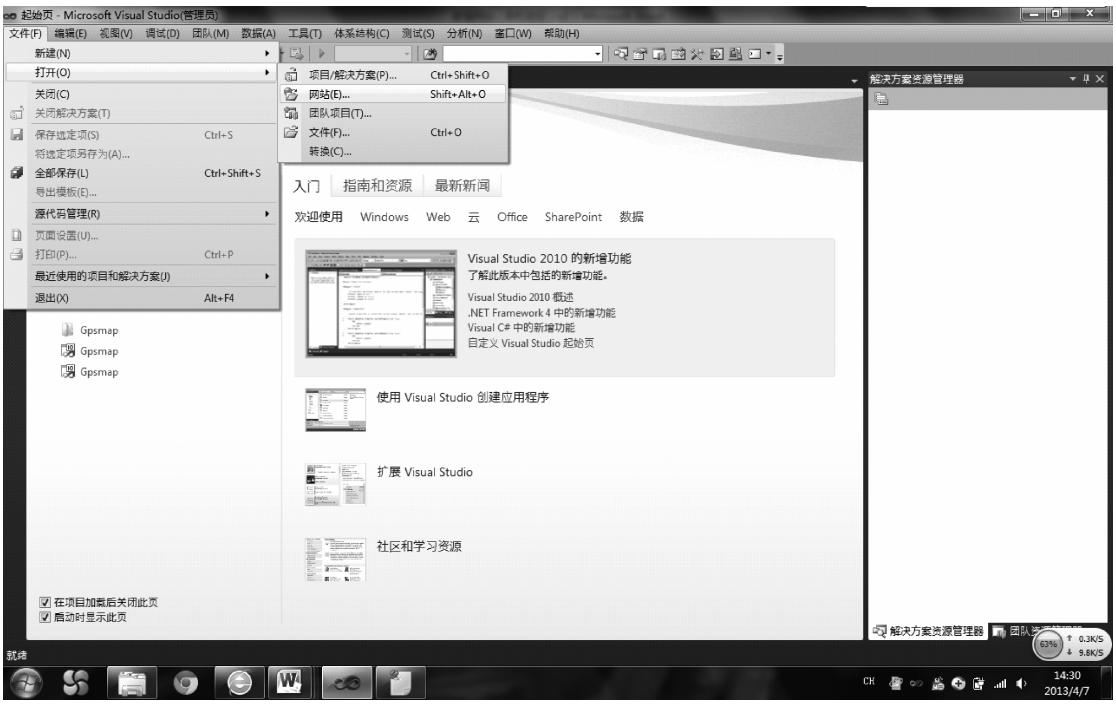


图 2-18　新建网站

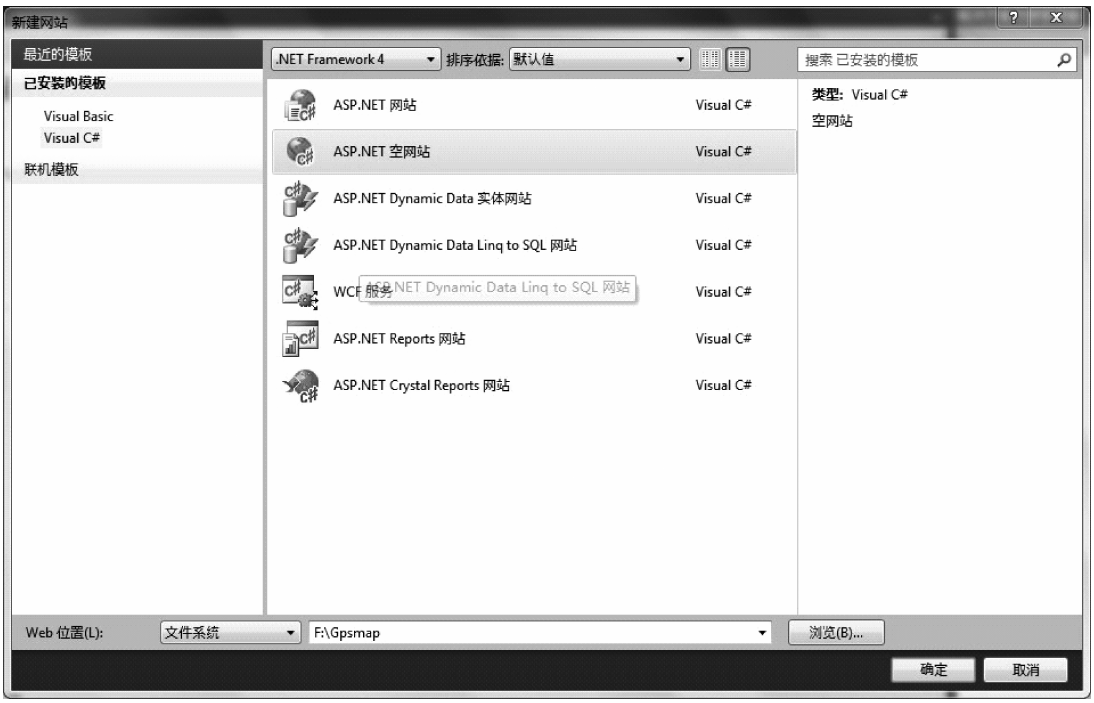


图 2-19　ASP. NET 空网站

本、CSS 样式、库文件等分别新建文件夹存储，如图 2-21 所示。

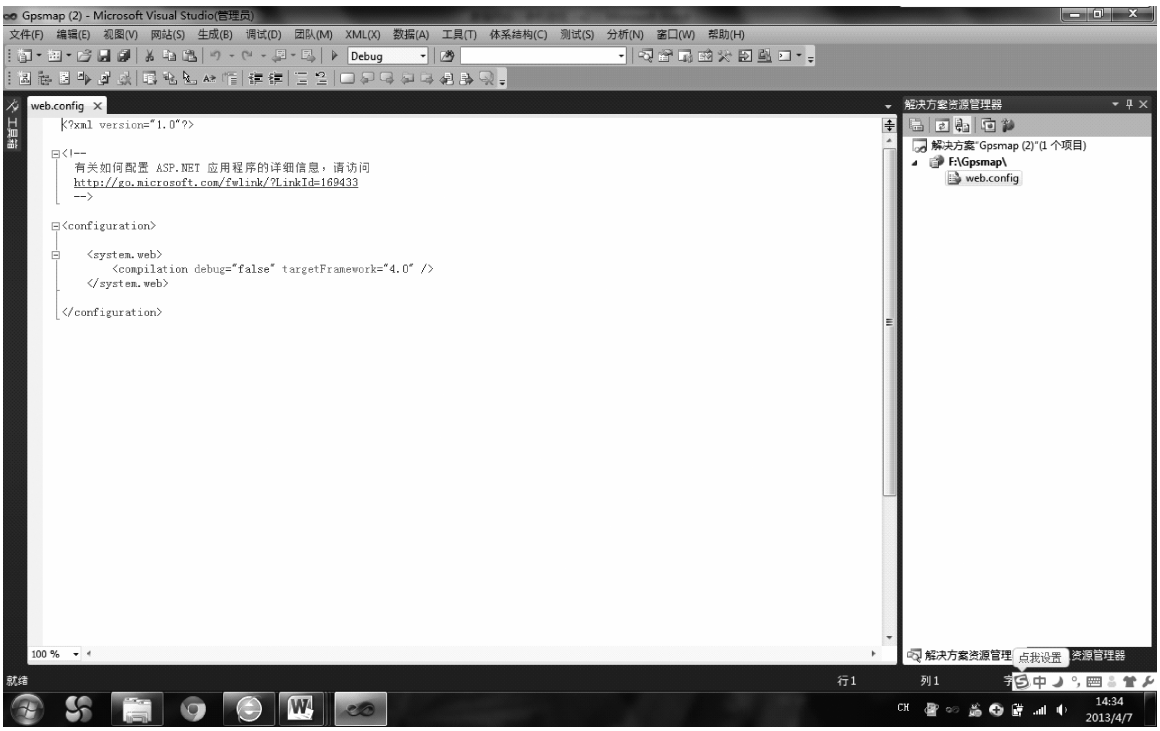


图 2-20　新建网站后显示界面

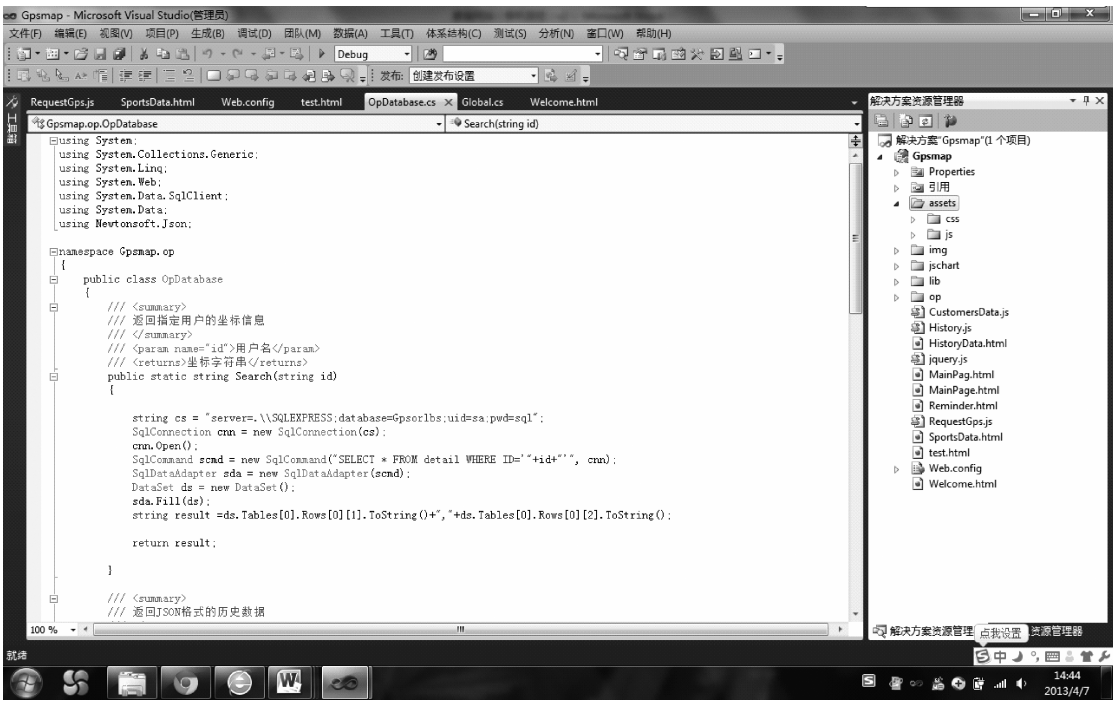


图 2-21　网站内容

2.6.5　发布网站

1）打开控制面板，选择管理工具，如图 2-22 所示。

图 2-22　进入管理工具界面

2）选择第一项 Internet 信息服务（IIS）管理器，如图 2-23 所示。

图 2-23　IIS 管理器界面

3）我们会看到左侧有网站目录，用鼠标右键单击“添加网站”，如图 2-24 所示。

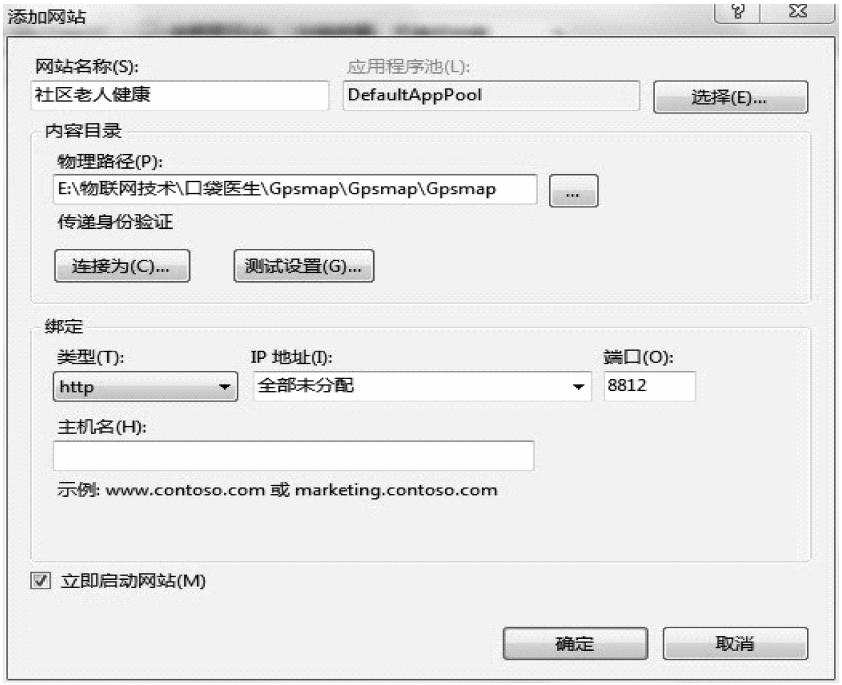


图 2-24　添加网站

按图 2-24 所示进行选择，起好网站名称，应用程序池可以选择默认，物理路径是 VS 里新建网站的路径，端口号可以从 1024 ~ 65535 之间任意选择，单击“确定”。如图 2-25 所示。左侧已经显示我们的网站了。

图 2-25　成功添加网站

4）这样在浏览器地址栏输入我们网站的地址“http：//localhost：8812/”就可以访问刚才新建的网站了，如图 2-26 所示。

图 2-26　在浏览器里访问网站

2.7　思考题

1）运行软件的时候遇到的问题都有哪些？分别是怎样解决的？

2）举例几种常见的网站漏洞。

2.8　实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤（包括软硬件设计方案、软件流程图、硬件原理图等）以及调试过程、心得体会，最后要附上程序，并按要求回答思考题 1）、2）。

参 考 文 献

[1]　刘海清，张永林. ASP 进行动态和交互式数据查询［J］. 计算机应用研究，2001（8）.

[2]　王志鹏，张海盛，沈益民，等. 基于 Web 的综合查询系统的设计与实现［J］. 计算机应用，1999（10）.

[3]　高深. 基于 Windows Server 2008 的网站服务器的构建［J］. 电子测试，2013（5）.

[4]　徐涛，范辉. 移动终端的定位跟踪系统研究与实现［J］. 无线互联科技，2013（5）.

[5]　吴诺. 都市社区居家养老服务体系中的信息化系统构建［J］. 价值工程，2012（13）.

第 3 章　视频安防监控系统

3.1　实验目的

1）了解智能家居安防监控系统的构成和功能。

2）了解智能家居远程视频监控方法。

3）熟悉一些图像分析方法，并通过使用 VS2010 和 OpenCV 学习几种用于图像处理的算法。

3.2　实验内容

1）通过网络摄像机对家庭进行远程实时监控。

2）配置 Visual Studio 和 OpenCV，使其满足图像处理的软件环境。

3）编写 C 语言程序，运用几种算法对图像进行分析处理。

3.3　实验所用仪表及设备

1）硬件：PC 一台、EasyN 的 F 系列网络摄像机、家庭网关、家庭网络服务器。

2）软件：Microsoft Visual Studio 2010、OpenCV2.3.1。

3.4　实验原理

3.4.1　安防监控系统概述

1. 安防系统三个基本要素

（1）探测

能及时地发现防范区域内的各种异常现象。安防系统探测的对象主要是人或人的行为。

（2）环境监控

安防系统要能对探测的结果做出评价（判断其真伪），要能控制事件发生的过程，了解事件处置的结果，必须具备监控的手段。

（3）周界管理和出入控制

安防系统要有明确的防范区域，即周（边）界。通过各种物理的、技术（电子）的手段，组成一个封闭的周界，并通过与周界结合的出入口，对人、物的出入进行管理和控制。安防系统应保证合法出入防范区域的人和物的通畅，有效地发现和阻止非法的出入。

2. 安防监控系统的功能

对于智能家居安防监控系统来说，能够实现防盗报警、消防报警、气体泄漏报警和紧急求助报警等功能。这些功能主要是靠安装在住宅周界和住宅内部各处的探测传感器将信号进行采集。由智能家居控制器进行智能判别，如果发现异常情况，则发出声光报警，同时将报警信号通过网络传送到主人的移动终端。

（1）家庭防盗报警功能

通过视频摄像头检测家中是否被盗，定时地传送相应的图像给中央控制器，由中央控制器分析图像并检测是否有人闯入，主人和小区警卫可通过安装在住户室内的报警控制器在小区管理中心得到信号从而快速接警处理。

（2）确保安全的生活空间

多层次监控系统确保安全的生活空间。家庭网络将监视触角延伸至家庭的每一个角落，智能安防实行安全防范系统自动化监控管理，提升家庭的安全保障。

（3）远程监控

系统将摄像机连接至中央控制系统，并进行远程控制，可随时通过计算机、手机远程查看各监控区域情况。

（4）及时了解家中的情况

监控设备可通过网络技术进行远程实时监视，远程录像查询回放及云台控制。利用先进的 3G 技术，在外用手机就可以查看家中实时画面，了解家中情况。用户可以在离开家的时候依然照顾和看护家里需要照顾的人。

（5）录像回查

按时间或事件检索案件发生现场的录像，进行嫌疑视频信息收集，搜索嫌疑人、嫌疑车辆等线索。对应系统功能需求：输出视频前端存储和后端备份存储；基于事件的视频存储；录像文件检索和回放等。

（6）视频智能分析和快速预警

对实时视频图像进行智能分析，及时发现可疑事件，并向家庭主人发出预警，同时切换出警情的图像进行自动跟踪。

3. 智能家居网络视频监控系统的结构和功能

基于安防系统（见图 3-1）的三要素和我们要实现的功能，智能家居网络视频监控系统主要由前端视频采集单元（PU）、传输网络（TU，包括视频数据传输单元、网络存储单元）、中心服务平台（CSP）和客户端单元（CU）四部分组成。系统在前端进行视频采集，经过压缩与优化算法等处理，通过网络传输进入中心服务平台，中心服务平台包含前端视频接入、视频分析与报警、平台管理、客户端接入等模块（见图 3-2）。客户端通过校园网或 IP 宽带、3G 等网络获取智能视频监控信息、进行预置等。

1）前端视频采集单元（PU），由摄像机、采集卡、数据压缩卡、云台控制机及其他外部设备组成，获取视频的原始数据，并进行初步的数据压缩。摄像机类型目前主要有数字视频录像机 DVR、网络摄像机 IPCAMERA 和网络摄像机 IPCAMERA（或模拟摄像机/视频服务器 IPENCODER）加 NVR 网络视频录像机几种形式；另外采用智能嵌入式芯片的视频采集单元主要是在前端集成了图像传感器、嵌入式处理器、图像处理电子卡、网络接口等。通过压缩优化算法和单一的背景差分算法等使摄像机实现实时的图像压缩、传输，并能跟踪目标产

生一定的反馈信号。

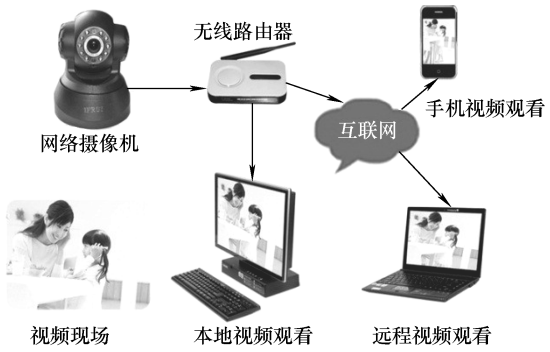


图 3-1 系统简化示意图

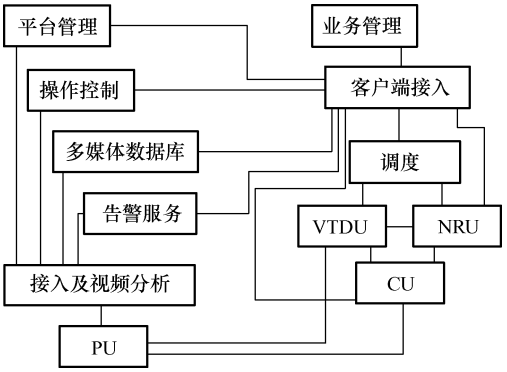


图 3-2 系统模块结构图

2）传输网络（TU），采用无线和有线的方式都可以实行。有线方式是互联网络平台，可支持各种互联网接入技术，如 ADSL、以太网、专线等，充分利用互联网的经济、方便的特性。无线方式是通过中国移动、中国联通、中国电信小灵猫等平台实施，网络覆盖率高。

3）中心服务平台（CSP），为认证、管理、运营、存储平台，支持多用户、多范围、多应用、多分布的整体监控，是联系各模块和客户端的枢纽。在智能家居中，中心服务平台的功能主要由家庭服务器完成。平台可被区分为以下七个功能区：

① 视频接入与分析模块。实现前端和平台间信令的交互，获取前段视频数据接入，分析与处理视频数据，并实现视频请求、接收、分发。视频分析包括运动目标检测、目标区域提取、目标识别等过程如图 3-3 所示。

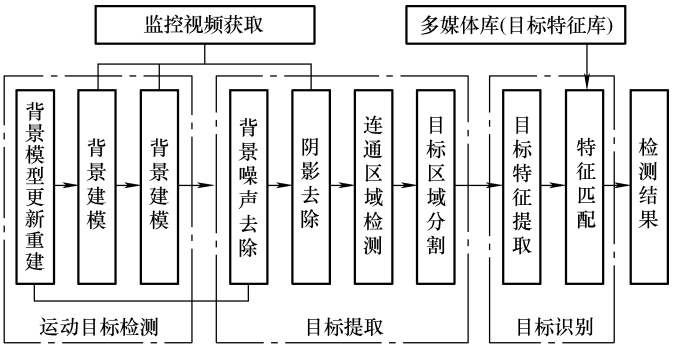


图 3-3 目标检测流程

② 预报警及联动发动模块。对视频的实时监控中触发预设阈值或产生异常产生声报警信号，并按报警级别发动联动措施，它是智能视频监控系统不可缺少的部分。

③ 多媒体数据库。存储待监控对象的多媒体特征数据，供识别与分析调用。

④ 用户认证。授权服务平台，存储用户信息，为用户授权接入、计费等。

⑤ 平台管理模块。用来实现对平台整体设备的管理，实现网管功能。

⑥ 客户端接入模块。为用户提供界面接入接口、实现客户端与平台间信令及信息的交互。操作控制模块来处理系统的各种控制信令，对客户端信令进行响应。

⑦ 调度模块。用来控制交换层设备、媒体流数据的流向和计费信息处理等。

4）客户端单元（CU），用户与中心服务平台交互的平台，获取远程监控实时视频信息、视频检索、控制摄像头和预置预报警阈值及转发告警信息。客户端支持多种形式的监控终端，用户可以通过手机、PDA、笔记本电脑等移动设备，在网络的任何一个接入点接入，

在通过用户认证后可以对现场监控，只要通过 IE 浏览器或客户端软件，就可控制远端设备，如控制摄像机的角度、镜头焦距等，并可接收报警信息。以智能手机为终端的客户端，基于 3G 或 Wi-Fi；以手机厂商提供的开发软件为平台，能实现用户任何时间、任何地点的视频监控；以 PC 为终端的客户端。基于多媒体技术，利用 PC 强大的处理能力，提供高质量的视频播放、智能识别等操作。其主要包含以下三个模块：

① 分析设置模块。分析设置模块是根据不同现场环境中的摄像头以其发挥的不同功能分别进行分析设置。首先是摄像头的定标，然后是告警预置。最后是目标行为分析参数设置。最终形成参数配置文件存储于中心服务器。

② 监控计划模块。对不同地段不同时间的同一路摄像头获取的视频可以实施多种视频监控方案。在场景监控条件发生改变或者监控目标需要临时调整时，可以设置各种监控计划，其包括一次性的、周期的、某一时间段的或默认计划等多种监控选择方案。

③ 实时视频播放模块。在客户端实现视频流的实时播放是一组过滤器和相关的应用程序来实现的，需要对从中心服务器以 RTP 转发过的视频压缩包进行还原，以一定视频压缩标准解码器进行解码，最后形成视频的实时播放。

4. 视频监控系统工作流程

系统以全数字网络视频监控技术为基础，利用网络资源优势，计算机软件技术、数字视频分析与处理等技术，完成分布式远程集中监控，实现资源共享、分级使用、权限管理、主动识别与跟踪、对象行为分析、联动预报警等系统。流程如下：

1）CU 通过客户端接入模块向平台管理模块请求访问某前端视频采集单元实时视频。

2）业务管理模块判断该 User 是否有访问权限、业务类型及设置了异地访问，另需判断 PU 是否本域。

3）若 PU 在异域，将请求发送给 PU 所在域 CSP。

4）PU 所在域 CSP 通过接入及视频分析模块、PU、VTDU 建立交互，形成媒体访问路由。并产生一个唤回号（Call ID）。

5）若路由终点为 PU，向 PU 发送 Call ID。

6）向路由上各 VTDU 发送媒体访问地址、Call ID。

7）PU 所在域 CSP 将确定的访问地址和 Call ID 返回给 CU 所在域 CSP。

8）CU 所在域 CSP 通过与 VTDU 交互确定本域内媒体访问路由。

9）向路由上各 VTDU 发送媒体访问地址、Call ID。

10）向 CU 返回媒体访问地址、Call ID。

11）各段分别请求建立媒体通道。

12）各被请求设备验证 Call ID 符合后，各段分别完成媒体通道建立，形成一条视频传输通道。

13）通道建立后，接入视频分析模块实时接收 PU 端视频数据，视频分析模块实时与多媒体数据库交互、鉴别是否有异常发生，如有，按照一定的算法分析识别视频对象，并分析对象行为，达到平台管理模块预置的警戒阈值时启动报警服务，然后通过客户端告知 CU。

3.4.2　硬件设计与实现

智能家居平台上视频安防的主要硬件为 EasyN 的 F 系列网络摄像机，用于 IP 网络基础

上的远程视频图像传输。其采用 MJPEG 硬件压缩技术，可以在 LAN/WAN 上以 30f/s 传输高质量（VGA 或 CIF）的实时视频图像。并且内置有 Web 服务器，支持 IE 浏览，可通过网络进行远程配置、启动或固件升级，设备管理和维护非常方便。

1. 设备的性能和功能

1）采用高性能、功能强大的媒体处理器 32bit RSIC。

2）高清晰度 COMS 感光芯片，每秒可达实时 25 帧。

3）采用优化 Motion-JPEG 视频压缩算法，窄带上实现高清图像传输。

4）最多可支持 5 个用户同时浏览，使用转发服务器功能则无人数限制。

5）内置 Web Server，方便用户使用标准的浏览器实现对现场的实时监控以及设置管理。

6）支持 Wi-Fi：IEEE 802.11b/g 无线网络。

7）支持远程系统升级。

8）支持动态域名解析，支持局域网和互联网（拨号上网、路由器上网等）。

9）支持单/双向语音对讲。

10）移动侦测报警功能，可设置灵敏度。

11）内置看门狗、异常自动恢复功能，网络中断后可自动连接。

12）动态报警功能，可以设置报警时间段。

13）具有发送邮件、FTP 图片、外部输出。

14）支持手机访问。

15）支持预置位，并支持报警时调用预置位。

2. 设备接口（见图 3-4 和图 3-5）

图 3-4　网络摄像头

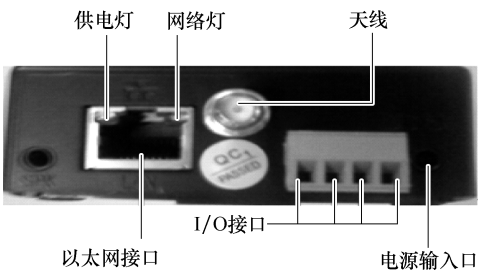


图 3-5　网络摄像头接口

供电灯：通电后常亮。

网络灯：通电后有数据传输时常闪。

天线：接收信号。

RJ45 以太网端口：用于通过网线将摄像机与 LAN 连接。

I/O 接口：报警输入，接 3、4 两个端子（输入端接地，低电平有效触发）；TTL 控制输出，接 1、2 两个端子。

电源输入口：接直流 5V 电源。

3. 设备规格和硬件参数表（见表 3-1）

表 3-1　设备规格表

系统特色	系统安全	三级账号、密码、用户多层权限管理
	自带动态 IP 域名系统（免费）	内嵌自主研发的 DDNS，终身免费专用域名使用，无需申请花生壳，彻底解决频繁掉线问题，连接速度快，如 http：//demo. ipcam. hk 机身编码为"demo"
	IE 多画面客户端（免费）	无需安装软件，IE 多画面监控、管理，短信报警设置，管理，报警图片保存功能
	集中监控系统（免费）	1，4，9，16，25，36，64 画面集中监控系统，无用户数量限制的专业版集中监控系统，集中监控，远程录像，远程回放录像，Skype 短信报警、Skype 电话报警、短信等多种报警方式
	手机访问	支持 iPhone、Windows Mobile、Symbian、安卓手机直接观看 支持具有 Java MIDP 2. 0 功能的手机
	微电脑处理器	32bit RSIC 嵌入式处理器
视频	压缩方式	Motion-JPEG-N
	信号系统	CMOS 30 万像素
	帧频率	25f/s
	分辨率	VGA（640 ×480），QVGA（320 ×240）
	图像调整	亮度，对比度，色度，饱和度（可调）
	白平衡，背光补偿	自动白平衡，自动背光补偿
网络	网络接口	RJ-45　10/100Mbit/s 自适以太网接口
	网络协议	支持 TCP/IP、HTTP、ICMP、DHCP、FTP、SMIP、PPPoE 等
	无线	Wi-Fi，IEEE 802. 11b/g（部分机型）
	在线用户数	直连最大观看人数 5 人，使用 P2P 技术用户在线观看不受限制
	支持 IP 地址	静态 IP 地址、动态 IP 地址、PPPoE 拨号
报警	输入/输出	1/1
	报警检测	移动侦测，检测灵敏度设置
	报警通知	支持 Email、FTP 图片上传、预置位调用、GPIO 控制等报警方式
证书	认证	ISO FCC CE

3. 4. 3　软件设计与实现

计算机视觉检测是建立在计算机视觉研究基础上的一门新兴检测技术，与计算机视觉研究的视觉模式识别、视觉理解等内容稍有不同，计算机视觉检测更侧重于获取被检测对象的精确参数，或者根据给出的参数提取出满足条件的目标。

计算机视觉检测过程主要有三个步骤：视觉影像的获取、视觉数据的处理、目标检测及结果输出。视觉影像的获取阶段完成检测对象的数字图像生成。不同的检测对象有时需采用不同的成像传感器。对可见光视觉检测，一般可采用 CCD 成像器件或 CMOS 成像器件。对不可见光视觉检测，还可以采用红外热像、X 光成像、超声成像以及雷达成像和声呐成像

等。各类成像传感器的使用，极大地拓展了计算机视觉检测的范围。视觉数据的处理阶段主要完成检测图像的图像增强、目标分割、特征提取等工作，这是一个由具体到抽象的过程。最后在第二步提取特征的基础上完成目标的识别及所需参数的测量，并输出结果。

智能视频监控系统就是基于计算机视觉检测技术的一个应用。其中，运动目标检测技术和目标识别与跟踪算法为该系统的核心技术。在实现目标检测前，首先要学习一些图像分析方法，例如帧差法、人脸检测、Canny 边缘检测和运动目标检测。

软件开发平台

针对视频监控系统所获取的图像，需要通过软件来进行分析和处理。本次实验系统的开发平台是 Microsoft Visual Studio 2010 和 OpenCV，并运用 C 语言实现了视频流中图像检测和分析的功能。

（1）VS2010

Visual Studio 是微软公司推出的开发环境，是目前较流行的 Windows 平台应用程序开发环境。Visual Studio 2010 版本于 2010 年 4 月 12 日上市，其集成开发环境（IDE）的界面被重新设计和组织，变得更加简单明了，如图 3-6 所示。Visual Studio 2010 同时带来了 NET Framework 4.0、Microsoft Visual Studio 2010 CTP（Community Technology Preview），并且支持开发面向 Windows 7 的应用程序。除了 Microsoft SQL Server，它还支持 IBM DB2 和 Oracle 数据库。Visual Studio 可以用来创建 Windows 平台下的 Windows 应用程序和网络应用程序，也可以用来创建网络服务、智能设备应用程序和 Office 插件。

图 3-6 Visual Studio 2010

（2）OpenCV（Open Source Computer Vision Library）

OpenCV 于 1999 年由 Intel 建立，如今由 Willow Garage 提供支持。OpenCV 是一个基于（开源）发行的跨平台计算机视觉库，可以运行在 Linux、Windows 和 Mac OS 等操作系统上。它轻量化而且高效——由一系列 C 函数和少量 C++类构成，同时提供了与 Python、Ruby、MATLAB 等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。OpenCV 拥有包括 300 多个 C 函数的跨平台的中、高层 API。它不依赖于其他的外部库——尽管也可以使用某些外部库。OpenCV 为 Intel® Integrated Performance Primitives（IPP）提供了透明接口，这意味着如果有为特定处理器优化的 IPP 库，则 OpenCV 将在运行时自动加载这些库。

3.4.4 图像处理算法

1. 运动目标检测

如果图像中的物体发生变化（位移或大小改变），那么该物体对应位置的灰度信息也会发生变化，而图像中没有发生变化或者变化微小的部分其灰度信息则不变或变化很小，这就是帧差法运动目标检测的基本原理。它能够较好地适应环境，具有简单，直接、容易实现的

特点。

2. 人脸检测

人脸检测是利用分析比较人脸视觉特征信息进行身份鉴别的计算机技术，其具有巨大的需求。因此需要有一个不错的基线技术使用，人脸检测是建立在最经常使用的分类器 Boosting 上，通过成千上万的物体各个角度的训练图像，训练出新的分类器。因此，对于此类识别任务，Haar 分类器是一个有用的工具。

OpenCV 中包含的一些检测器实现了人脸检测。它首先是由 Paul Viola 和 Michael Jones 设计的，称为 Viola-Jones 检测器，并由 Rainer Lienhart 和 Jochen Maydt 用对角特征扩展。OpenCV 称这个检测器为“Haar 分类器”是因为它使用 Haar 特征或更准确地描述是类 Haar 的小波特征，该特征由矩形图像区域的加减组成。除了人脸外，训练方法（createsamples（）和 haartraining（））和识别方法（cvHaarDetectObjects（））可以用于具有纹理的近似刚性的任何物体。

OpenCV 中预先训练好的物体检测器位于…/opencv/data/haarcascades，其中正面人脸识别效果最好的模型文件为 haarcascade_ frontalface_ alt2. xml。Haar 分类器是一个监督分类器，先对图像进行直方图均衡化并归一化到同样大小，然后标记里面是否包含要检测的物体，大多数情况下要检测的物体是人脸。

Viola-Jones 识别器使用 AdaBoost，但是把它组织为筛选式的级联分类器，每个节点都是多个树构成的分类器，且每个节点的正确识别率均很高，但是正确拒绝率很低。在任一级计算中，如果一旦获得目标“不在类别中”的结论，则计算终止，算法也宣布该位置上没有人脸。因此只有通过分类器中的所有级别，才会认为物体被检测到。这样当目标出现概率较低的时候，筛选式的级联分类器可以显著地降低计算量，因为大部分被检测的区域都可以很早被筛选掉，迅速判断出此处无人脸。

Viola-Jones 的筛选式级分类器中，弱分类器是一个多数情况下只有一层的决策树。一层决策树允许下面形式的决策：判断特征 f 的值 v 大于某个阈值 t；Yes 表示可能是人脸，No 表示不是人脸：

$$f_i = \begin{cases} +1 v_i \geqslant t_i \\ -1 v_i < t_i \end{cases}$$

训练中，Viola-Jones 分类器在每个弱分类器中使用的类 Haar-like 特征个数可以设置。但是大多数情况下，使用 1 个特征，最多用 3 个。然后提升算法迭代地建立一个由那些弱分类器的加权和组成的强分类器。Viola-Jones 使用如下分类函数：

$$F = \text{sign}(w_1 f_1 + w_2 f_2 + \cdots + w_n f_n)$$

如果加权和小于 0，则符号函数返回 -1；加权和为 0，则返回 0；大于 0，则返回 1。在第一次遍历数据的时候，训练得到 f_1 的阈值 t_1，它最好地区分了输入数据。然后 Boosting 算法使用得到的错误来计算投票权值 w_1。根据传统的 AdaBoost，每个特征向量将根据是否被正确分类，重新赋予或高或低的权值。一旦一个节点训练完成，剩余的数据将被用来训练下一个节点，以此类推。

3. Canny 边缘检测

Canny 边缘检测是图像分割的一种，图像分割是指根据灰度、颜色、纹理、几何形状等信息把图像分成各具特点、互不重叠的区域并提取出感兴趣目标的技术和过程。简单地讲，

就是在一幅图像中，把目标从背景中分离出来，以便对感兴趣目标进行更进一步的分析。图像分割是由图像处理到图像分析的关键步骤，是对图像进行分析和理解的基本前提。

Canny 边缘检测是高斯函数的一阶导数，是对信噪比与定位之乘积的最优化逼近算子。Canny 边缘检测具有 3 个特点：低概率的错标非边缘点和低概率不标真实边缘点；检测出来的边缘点应该尽可能地靠近真实边缘中心；边缘响应是单值的。

Canny 算法的实现步骤：

Step1：用高斯滤波器平滑图像，去除图像噪声。一般选择方差为 1.4 的高斯函数模板和图像进行卷积运算。

Step2：用一阶偏导的有限差分来计算梯度的幅值和方向。使用梯度算子计算 x 和 y 方向的偏导数、方向角和梯度幅值。

Step3：对梯度幅值应用非极大值抑制。幅值 M 越大，其对应的图像梯度值也越大，但这还不足以确定边缘，因为这里仅把图像快速变化的问题转化成求幅值局部最大值问题，为确定边缘，必须细化幅值图像中的屋脊带，只保留幅值局部变化最大的点，生成细化的边缘。

Step4：用双阈值算法检测并且连接边缘。双阈值法使 Canny 算子提取的边缘点更具有鲁棒性，高低阈值分别表示为 Hth 和 Lth，对于高阈值 Hth 的选取，基于计算出的图像梯度值对应的直方图进行选取。在一幅图像中，非边缘点数目在总图像像素点数目中占的比例表示为 Hratio，根据图像梯度值对应的直方图累加，累加数目达到总像素数目的 Hratio 时，对应的图像梯度值设置为 Hth，在文中设定 Hratio 为 0.7。低阈值 Lth 的选择通过 Lth = Lratio * Hth 得到，文中 Lratio 设定为 0.4。最后通过对边缘点的标记和领域关系进行连接得到最后的边缘检测图。

3.5 实验步骤

3.5.1 访问网络摄像机对家庭进行远程监控

1）打开 Web 页面，在网页地址栏里输入下列地址访问网络摄像头：

IP 地址http：//115.25.48.151：81

http：//115.25.48.151：83

http：//115.25.48.151：82

http：//115.25.48.151：84

2）输入相应登录信息，单击“确定”按钮即可登录。默认登录用户名为 admin，密码无，如图 3-7 所示。

3）登录后显示如图 3-8 所示界面。

4）选择 IE 浏览器登录。登录后会出现网络摄像机界面，如图 3-9 所示，可通过单击界面的按钮来控制摄像头，观察家内状况。

3.5.2 配置 VS2010 和 OpenCV

开发采用的是 C 语言，需要配置 OpenCV，目前最新版本是 2.4.4。

图 3-7　登录界面

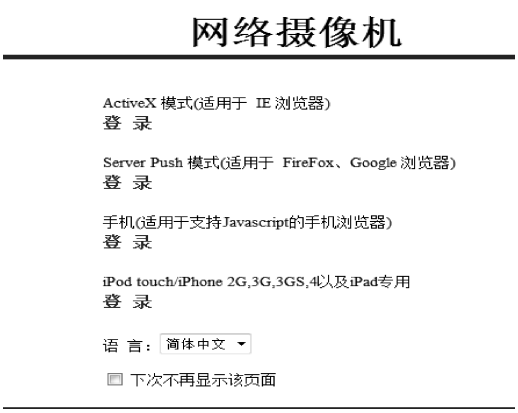


图 3-8　登录后界面 1

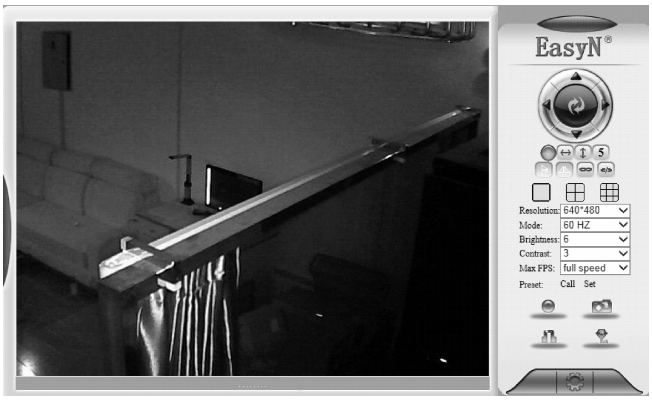


图 3-9　登录后界面 2

1）首先设置 VS，配置 Visual C ++ 开发设置，如图 3-10 所示。

图 3-10　VS 设置集合

2）使用 OpenCV 首先需要进行环境变量调配，在环境变量中变量名 Path 中加入 OpenCV 对应的路径，如图 3-11 所示。

图 3-11　OpenCV 环境变量

下面为具体操作顺序，逐步单击选择以下选项

计算机→(右键）属性→高级系统设置→高级（标签)→环境变量→(双击）path（用户，系统里面的 path 任选其一)→在变量值里面添加“opencv\build\x86\vc10\bin”和“opencv\build\common\tbb\ia32\vc10”

3）其次要在所创建工程中属性 VC ++ 目录中包含目录和库目录中加入 OpenCV 中相应的目录：

下面为具体操作顺序，逐步单击选择以下选项

项目（菜单项)→…属性→VC ++ 目录：需要配置“包含目录”和“库目录”两项。每次新建工程都要重新配置，如图 3-12 和图 3-13所示。

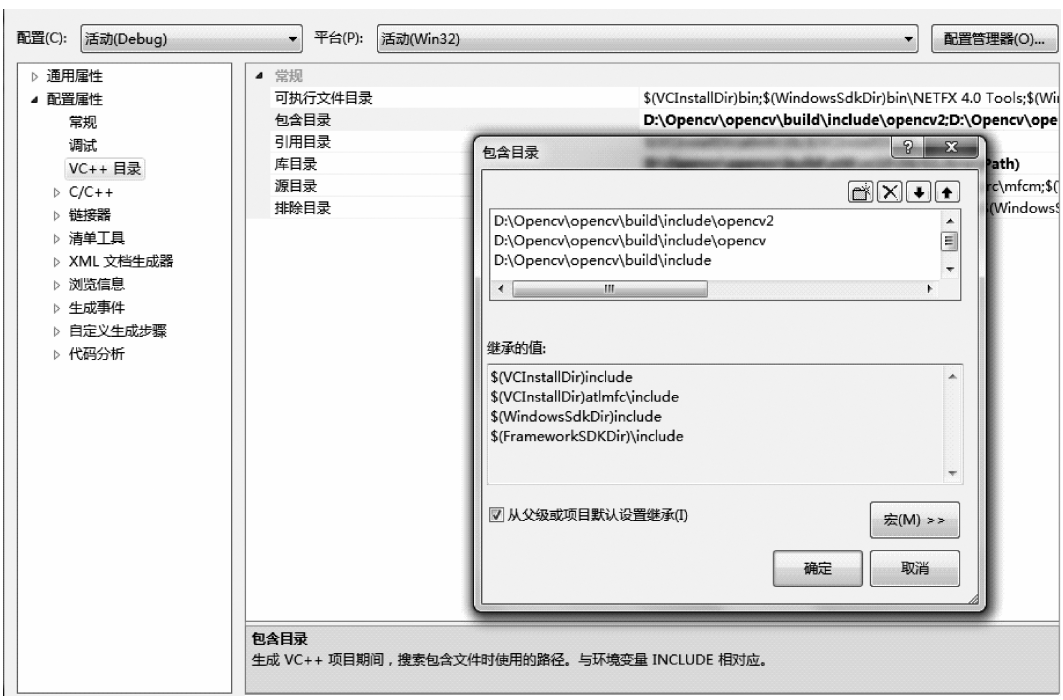


图 3-12　OpenCV 工程目录 1

4）最后要在链接器下的输入项中的附加依赖项中加入相应版本的依赖项，如图 3-14 所示。

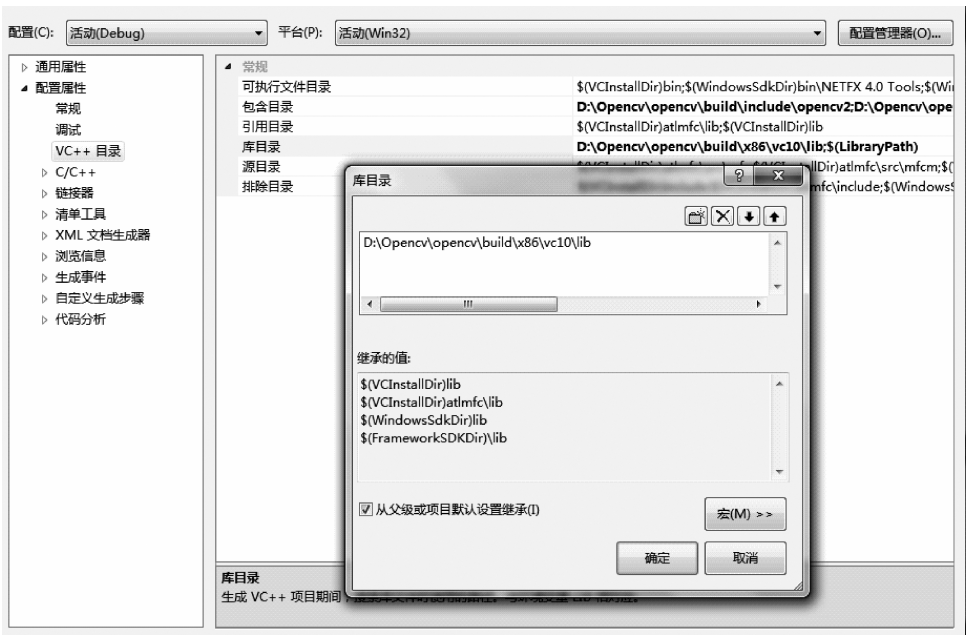


图 3-13　OpenCV 工程目录 2

下面为具体操作顺序，逐步单击选择以下选项

项目（菜单项)→…属性→连接器→输入→附加依赖项

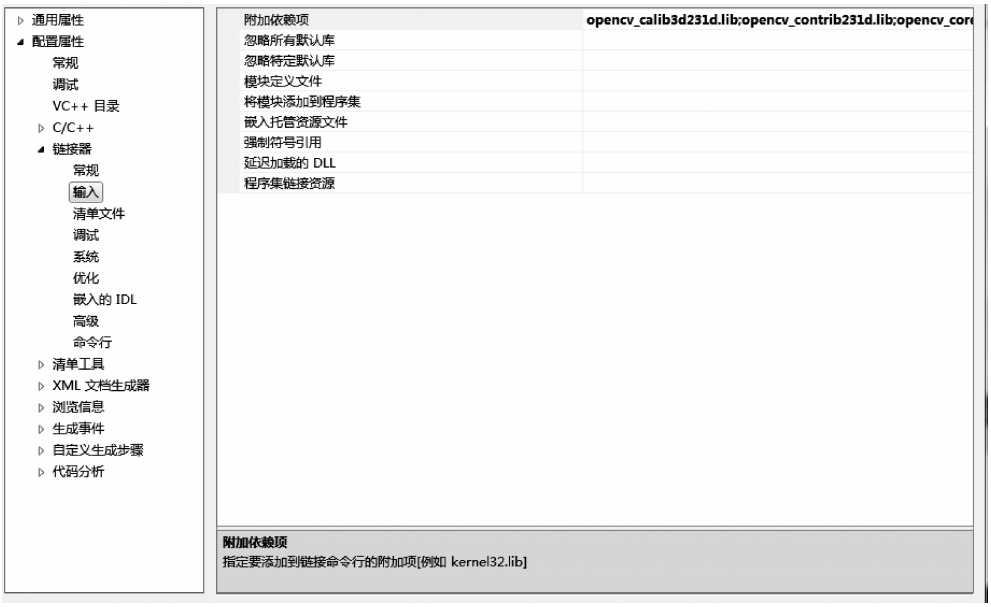


图 3-14　OpenCV 依赖项

5）针对 Debug 配置添加以下库：

opencv_calib3d240d. lib

opencv_contrib240d. lib

opencv_core240d. lib
opencv_features2d240d. lib
opencv_flann240d. lib
opencv_gpu240d. lib
opencv_highgui240d. lib
opencv_imgproc240d. lib
opencv_legacy240d. lib
opencv_ml240d. lib
opencv_objdetect240d. lib
opencv_ts240d. lib
opencv_video240d. lib
完成所有配置后就能进行系统的开发，以及系统的运行。

6）用 VS2010 新建控制台工程测试。

下面为具体操作顺序，逐步单击选择以下选项

文件→新建→项目→Visual C + +→Win32 控制台应用程序（输入名称 test），如图 3-15 所示。

图 3-15　新建工程 1

下面为具体操作顺序，逐步单击选择以下选项

确定→下一步→附加选项选“空项目”→完成，如图 3-16 所示。

下面为具体操作顺序，逐步单击选择以下选项

解决方案资源管理器→源文件（右键）→添加→新建项，如图 3-17 所示。

下面为具体操作顺序，逐步单击选择以下选项

Visual C + +→C + +文件：输入名称 test，单击“添加”按钮，如图 3-18 所示。

图 3-16　新建工程 2

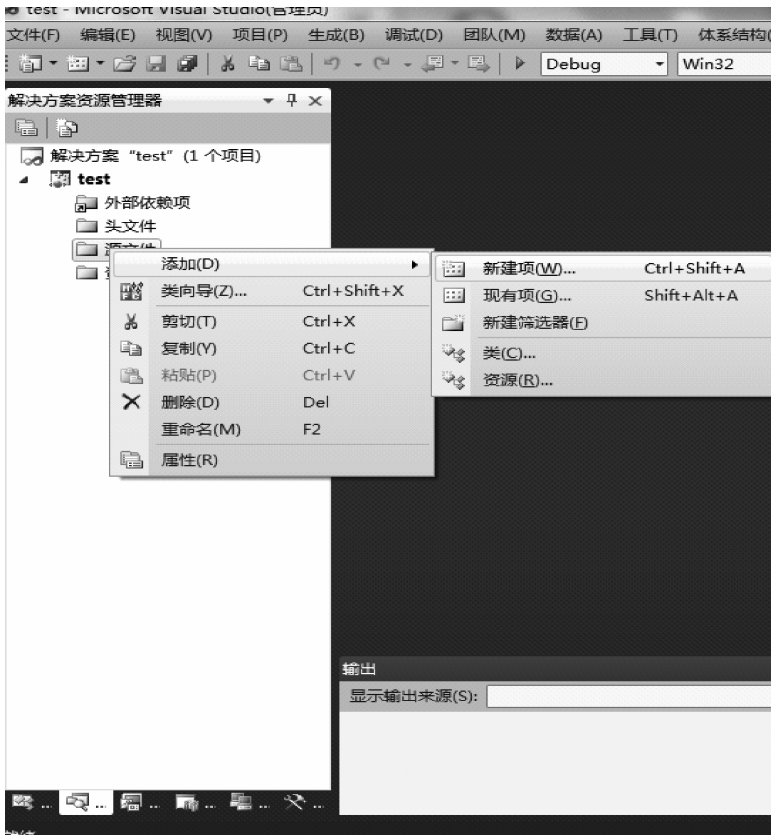


图 3-17　新建项

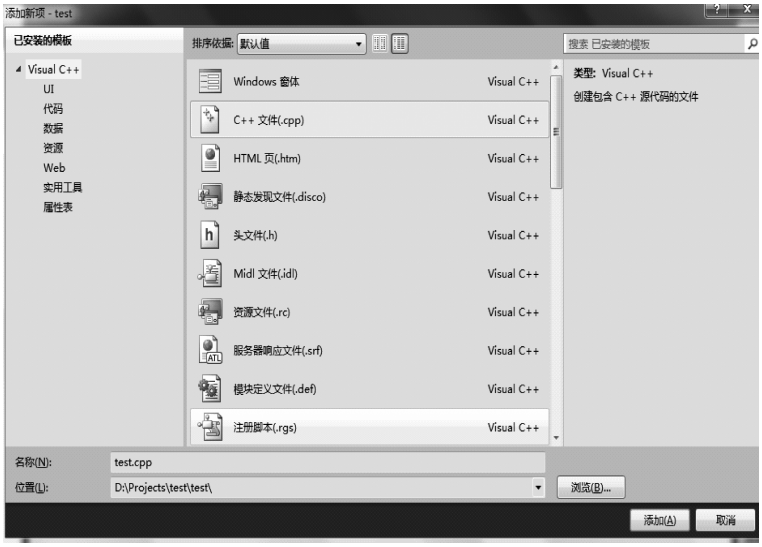


图 3-18 新建文件

3.5.3 运动目标检测

帧差法是背景差图法中的一种，只不过是帧差法不需要建模，因为它的背景模型就是上一帧的图，所以速度非常快。本程序利用 C 语言调用 OpenCV 中的相关图像函数来实现图像的实时检测。此次程序实现的是对视频中图像的检测，由于摄像头的最主要的功能是安防监控，根据图像间做差的方法可以实时地检测家中是否有可疑的人物出现，是否有东西破损或者丢失。本程序设定 3 个显示的窗口，以得到对视频的检测。video1 中显示的是上一时间的图像，video2 中显示的是当前的视频图像，video3 中显示的是这两幅图像的差的图像。每间隔一段时间就把新抓取到的图像传到各个窗口进行做差显示，这个时间可以根据需要自己设定其时间间隔的长短，从而达到自己的需求。

1）在 VS2010 中运行根据帧差法配置好的工程。

2）在程序中修改本地视频的地址 capture = cvCreateFileCapture(" e: \a. avi")。

3）尝试不同的时间间隔来对比程序的效果 cvWaitKey （1000）。

4）也可尝试其他视频进行对比，观察三个窗口的图像区别，并做记录和分析。

图 3-19 为所获取的图像效果。

图 3-19 帧差法

以下为程序代码：

```
#include "stdafx.h"
#include "cv.h"
#include "highgui.h"
#include <stdio.h>
# define NULL 0
int main()
{
IplImage *res=0;
IplImage *res0=0;
IplImage *image=0;
IplImage *image1=0;
CvCapture *capture=NULL;

capture=cvCreateFileCapture("e:\a.avi");
IplImage *frame=0;
frame=cvQueryFrame(capture);
int image_width=(int)frame→width;
int image_height=(int)frame→height;

IplImage *res_flip=NULL;
IplImage *Gimage1_flip=NULL;
IplImage *Gimage2_flip=NULL;

image=cvCreateImage(cvSize(image_width,image_height),IPL_DEPTH_8U,3);
image1=cvCreateImage(cvSize(image_width,image_height),IPL_DEPTH_8U,3);
IplImage *image2=cvCreateImage(cvSize(image_width,image_height),IPL_DEPTH_8U,3);
res=cvCreateImage(cvSize(image_width,image_height),IPL_DEPTH_8U,1);
res0=cvCreateImage(cvSize(image_width,image_height),IPL_DEPTH_8U,3);
cvNamedWindow("video1",1);
cvNamedWindow("video2",1);
cvNamedWindow("video3",1);
while(image1)
{
 image=cvQueryFrame(capture);
 if(image==NULL)
 {
 break;
}
```

```
    cvCopy(image,image1,NULL);
    image = cvQueryFrame(capture);
    cvCopy(image,image2,NULL);

    cvAbsDiff(image1,image2,res0);
    cvCvtColor(res0,res,CV_RGB2GRAY);

    cvAdaptiveThreshold(res,res,255,CV_ADAPTIVE_THRESH_MEAN_C,
    CV_THRESH_BINARY_INV,5,5);

    cvShowImage("video1",image1);
    cvShowImage("video2",image2);
    cvShowImage("video3",res);
    cvWaitKey(1000);
}
    cvDestroyWindow("video3");
    cvDestroyWindow("video2");
    cvDestroyWindow("video1");

    cvReleaseImage(&res);
    cvReleaseImage(&res0);
    cvReleaseImage(&image2);
    cvReleaseImage(&image1);
    cvReleaseImage(&image);

    cvReleaseCapture(&capture);
    return 0;
}
```

5）具体程序分析如下。

对视频的分析第一步就是对视频的调用，以下程序即是对所要分析视频的输入调用：

```
capture = cvCreateFileCapture("C:\\Users\\fengzi\\Desktop\\5.mov");
IplImage *frame=0;
frame = cvQueryFrame(capture);
int image_width = (int)frame→width;
int image_height = (int)frame→height;
```

/*上面这一段表示提取视频输入*/,"C:\\Users\\fengzi\\Desktop\\5.mov" 代表视频所在的路径。

对所要抓取的图像进行创建和内存的预留，并建立3个显示窗口。具体程序如下：

```
Image = cvCreateImage(cvSize(image_width,image_height),IPL_DEPTH_8U,3);
```

```
image1 = cvCreateImage( cvSize( image_width, image_height ), IPL_DEPTH_8U, 3 );
IplImage  * image2 = cvCreateImage( cvSize( image_width, image_height ), IPL_DEPTH_8U, 3 );
res = cvCreateImage( cvSize( image_width, image_height ), IPL_DEPTH_8U, 1 );
res0 = cvCreateImage( cvSize( image_width, image_height ), IPL_DEPTH_8U, 3 );
cvNamedWindow( "video1", 1 );
cvNamedWindow( "video2", 1 );
cvNamedWindow( "video3", 1 );
```

抓取图像并传到预先分配的内存中：

```
image = cvQueryFrame( capture );
cvCopy( image, image1, NULL );
image = cvQueryFrame( capture );
cvCopy( image, image2, NULL );
```

对抓取到的两个图像进行做差：

```
cvAbsDiff( image1, image2, res0 );
cvCvtColor( res0, res, CV_RGB2GRAY );
```

对结果进行显示：

```
cvShowImage( "video1", image1 );
cvShowImage( "video2", image2 );
cvShowImage( "video3", res );
```

3.5.4　人脸检测

1. 网络摄像头人脸检测

可以通过网络摄像头从 FTTP 获取图像，然后运用人脸检测对获取的图像进行处理，从而实现对人脸的检测。系统仍然采用在 VS2010 下利用 OpenCV 库的方法进行开发，同时利用 VS2010 中 MFC 类库设计界面程序。

1）在 VS2010 中运行已经配置好的工程。

2）在程序中填写所要监控摄像头的地址和抓取图像保存的位置，下面为需要修改的程序位置。

```
http→Init( NULL, "http://115.25.48.151:82/snapshot.cgi? user = admin&pwd = ",
"E:\\2.jpg" );
```

3）运行程序之后获取如图 3-20 所示界面。

图像获取采用封装好的 HttpDownload 类，调用类中的 Init 就可以获取图像保存到相应的路径。

关键代码：

```
CHttpDownload *  http = new CHttpDownload( );
http→Init( NULL, " ", "" );
http→Start( );
SetTimer( 1, 1000, NULL );
```

利用 Settimer 可以响应相应的 Ontimer 函数让图片显示在界面中。

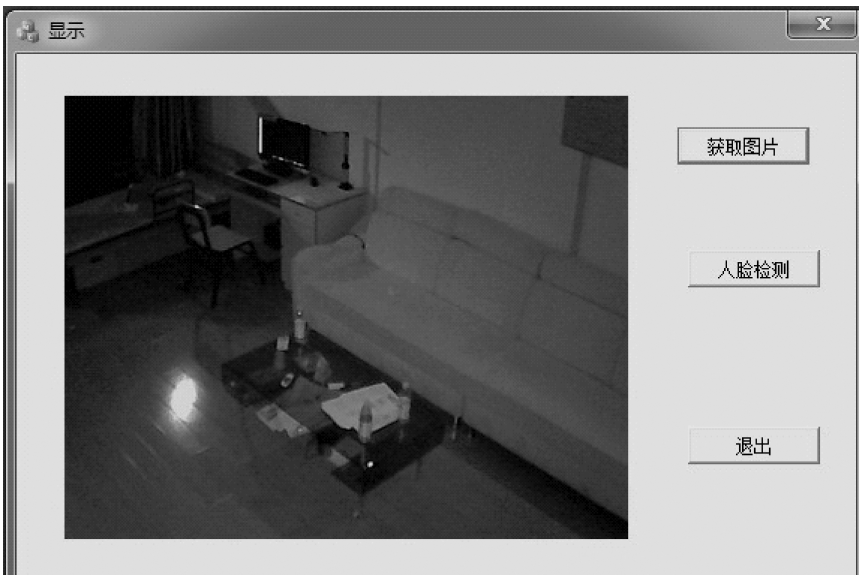


图 3-20 网络摄像头截图

4）单击获取图片会保存实时监控的图像。

5）如果图像中有人脸，单击人脸检测将会获取人脸识别。

以下为关键程序代码：

```
void CElevenDlg::OnBnClickedButton1()
{
    CHttpDownload * http = new CHttpDownload();
    http→Init(NULL,"http://115.25.48.151:82/snapshot.cgi? user=admin&pwd=",
"E:\\2.jpg");
    http→Start();
    SetTimer(1,1000,NULL);
}
void CElevenDlg::OnTimer(UINT_PTR nIDEvent)
{
    if(nIDEvent==1)
    {
        IplImage * img=cvLoadImage("E:\\2.jpg",1);
        m_CvvImage.CopyOf(img);
        m_CvvImage.DrawToHDC(m_pPicCtlHdc,m_PicCtlRect);
        KillTimer(1);
    }
    CDialogEx::OnTimer(nIDEvent);
}
void CElevenDlg::OnBnClickedButton2()
```

```
{
        CvHaarClassifierCascade * cascade =
(CvHaarClassifierCascade * ) cvLoad(cascade_name,0,0,0);
    storage = cvCreateMemStorage(0);
    IplImage * img = cvLoadImage("E:\\2.jpg",1);
    if(img)
    {
        static CvScalar colors[] =
        {
            {{0,0,255}},
            {{0,128,255}},
            {{0,255,255}},
            {{0,255,0}},
            {{255,128,0}},
            {{255,255,0}},
            {{255,0,0}},
            {{255,0,255}}
        };
        double scale = 1.3;
        IplImage * gray = cvCreateImage(cvSize(img→width,img→height),8,1);
        IplImage * small_img = cvCreateImage(cvSize(cvRound(img→width/scale),cvRound
(img→height/scale)),8,1);

        cvCvtColor(img,gray,CV_BGR2GRAY);
        cvResize(gray,small_img,CV_INTER_LINEAR);
        cvEqualizeHist(small_img,small_img);
        cvClearMemStorage(storage);

        if(cascade)
        {
        CvSeq * faces = cvHaarDetectObjects(small_img,cascade,storage,1.1,2,0,
                                            cvSize(30,30));
            for(int i = 0; i < (faces ? faces→total : 0); i + + )
            {
                CvRect * r = (CvRect * ) cvGetSeqElem(faces,i);
                CvPoint center;
                int radius;
                center.x = cvRound((r→x + r→width * 0.5) * scale);
                center.y = cvRound((r→y + r→height * 0.5) * scale);
```

```
                radius = cvRound((r→width + r→height) * 0.25 * scale);
                cvCircle(img,center,radius,colors[i%8],3,8,0);
            }
        }
        m_CvvImage.CopyOf(img);
        m_CvvImage.DrawToHDC(m_pPicCtlHdc,m_PicCtlRect);

    }
}
```

2. 本地视频人脸检测

1）在 VS2010 中运行已经配置好的工程，并且在程序中设置需要人脸识别的视频位置，下面为需要修改的程序位置。

```
CvCapture * capture = cvCreateFileCapture("E:\\face2.mov");
```

2）程序运行后获得如图 3-21 所示。

3）尝试其他视频进行对比。

图 3-21 本地视频

3.5.5 基于阈值分割的边缘检测

阈值分割法是一种基于区域的图像分割技术。图像阈值分割是一种传统的最常用的图像分割方法，因其实现简单、计算量小、性能较稳定而成为图像分割中最基本和应用最广泛的分割技术。它特别适用于目标和背景占据不同灰度级范围的图像。它不仅可以极大地压缩数据量，而且也大大简化了分析和处理步骤，因此在很多情况下，是进行图像分析、特征提取与模式识别之前必要的图像预处理过程。图像阈值化的目的是要按照灰度级，对像素集合进行一个划分，得到的每个子集形成一个与现实景物相对应的区域，各个区域内部具有一致的属性，而相邻区域布局也有这种一致属性。这样的划分可以通过从灰度级出发选取一个或多个阈值来实现。

基本原理是：通过设定不同的特征阈值，把图像像素点分为若干类。

常用的特征包括：直接来自原始图像的灰度或彩色特征；由原始灰度或彩色值变换得到的特征。设原始图像为 $f(x, y)$，按照一定的准则在 $f(x, y)$ 中找到特征值 T，将图像分割为两个部分，分割后的图像为

若取：b0 = 0（黑），b1 = 1（白），即为我们通常所说的图像二值化。

1）单击运行 VS2010 中根据阈值分割配置好的工程。

2）在程序中修改本地图像地址 image = cvLoadImage("E:\\open.jpg")。

3）运行程序后会获得如图 3-22 所示两个窗口，调节不同的阈值可获得不同的边缘检测效果。

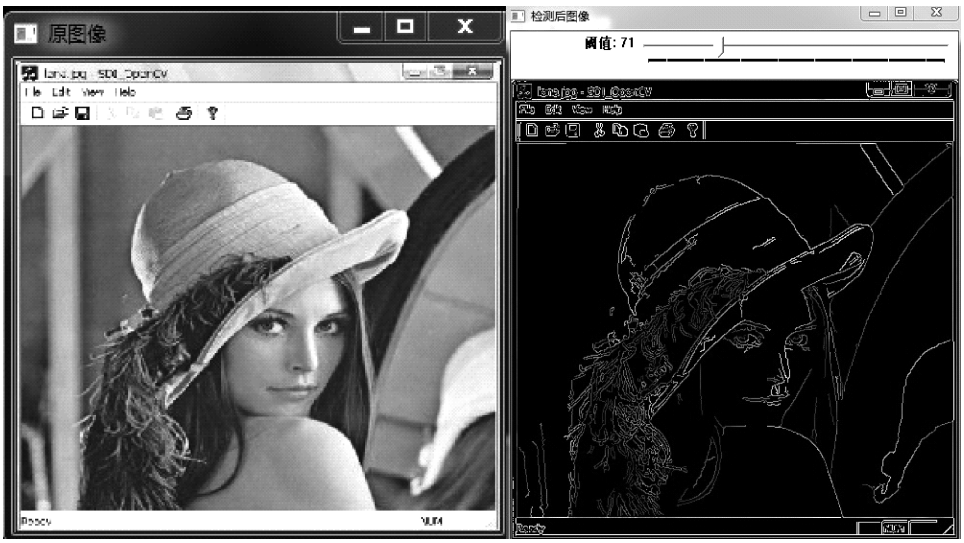


图 3-22　边缘检测

图像中出现的可调节滚动条是由以下函数来实现的：

```
void on_trackbar(int i)
{
    cvSmooth(gray,edge,CV_BLUR,3,3,0,0);
    cvCanny(gray,edge,(float)edge_thresh,(float)edge_thresh*3,3);
    cvZero(cedge);
    cvCopy(image,cedge,edge);
    cvShowImage(wndname,cedge);
}
```

该函数是一个 trackbar 滚动条回调函数，通过将灰度化的图像读入后对图像每个像素邻域求和，做尺度变换，然后用 Canny 边缘算子对图像进行边缘检测。用滚动条控制阈值的选定，阈值设定越高，图像中边缘轮廓越少。

关键代码：

```
#include "stdafx.h"
#include"highgui.h"
#include"cxcore.h"
#include"cv.h"
char wndname[] = "检测后图像";
char tbarname[] = "阈值";
int edge_thresh = 1;
IplImage  *image, *cedge, *gray, *edge;
void on_trackbar(int i)
{
    cvSmooth(gray,edge,CV_BLUR,3,3,0,0);
```

```
    cvCanny(gray,edge,(float)edge_thresh,(float)edge_thresh*3,3);
    cvZero(cedge);
    cvCopy(image,cedge,edge);
    cvShowImage(wndname,cedge);
}
int main()
{
    image=cvLoadImage("E:\\open.jpg");
    if(!image)
    {
        return-1;
    }
    cedge=cvCreateImage(cvSize(image→width,image→height),IPL_DEPTH_8U,3);
    gray=cvCreateImage(cvSize(image→width,image→height),IPL_DEPTH_8U,1);
    edge=cvCreateImage(cvSize(image→width,image→height),IPL_DEPTH_8U,1);
    cvCvtColor(image,gray,CV_BGR2GRAY);
    cvNamedWindow(wndname,1);
    cvNamedWindow("原图像",2);
    cvShowImage("原图像",image);
    cvCreateTrackbar(tbarname,wndname,&edge_thresh,300,on_trackbar);
    on_trackbar(0);
    cvWaitKey(0);
    cvReleaseImage(&image);
    cvDestroyWindow(wndname);
    cvDestroyWindow("原图像");
    return 0;
}
```

3.6 思考题

1）视频监控系统都有哪些模块？简略描述各模块的作用。

2）观察运用帧差法时三个窗口中图像的区别，分析产生不同图像的原因。

3）在使用阈值分割边缘检测时，为什么阈值越高图像中边缘轮廓越少？

3.7 实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤（包括软硬件设计方案、软件流程图、硬件原理图等）以及调试过程、心得体会，最后要附上程序，并按要求回答思考题1）、2）、3）。

参 考 文 献

[1] 陈凤东，洪炳镕. 基于动态阈值背景差分算法的目标检测方法［J］. 哈尔滨工业大学. 2005，37（7）.

[2] 黄文丽，范勇，李绘卓，等. 改进的混合高斯算法［J］. 计算机工程与设计. 2011，32（2）.

[3] 王辉. 背景差分图像处理［D］. 哈尔滨：哈尔滨理工大学，2005.

[4] 李泽峰，朱凌云. 基于改进背景差法的运动目标检测［J］. 仪表技术. 2012（1）.

[5] 安博文，艾燕. 实时图像的运动目标检测及跟踪［J］. 计算机仿真. 2012，29（2）.

[6] 樊晓亮，杨晋吉. 基于帧间差分的背景提取与更新算法［J］. 计算机工程. 2011，37（22）.

[7] 薛丽霞，罗艳丽，王佐成. 基于帧间差分的自适应运动目标检测方法［J］. 计算机应用研究. 2011，28（4）.

[8] 张咏，李太君，李枚芳. 利用改进的背景差法进行运动目标检测［J］. 现代电子技术. 2012，35（8）.

[9] 黄鑫娟，周洁敏，刘伯扬. 自适应混合高斯背景模型的运动目标检测方法［J］. 计算机应用. 2010，30（1）.

[10] 王小峰，王吉华，王远鹏，等. 混合高斯模型和差分法相融合的运动目标检测［J］. 小型微型计算机系统. 2013，34（3）.

[11] 侯宏录，李宁鸟. 一种基于运动目标识别的智能视频监控系统［J］. 兵工自动化. 2012，31（3）.

[12] 杨红军. 智能视频监控系统的设计研究［J］. 科技情报开发与经济. 2010，20（4）.

[13] 范亚男，葛卫丽. 智能视频监控系统发展及应用［J］. 价值工程. 2010，29（17）.

[14] 蔺光逢，范引娣. 智能视频监控中的人体检测与识别［J］. 计算机应用. 2007，27（z2）.

[15] 於时才，张文香，龚仁彬. 智能视频监控中的一种快速运动目标检测方法［J］. 兰州理工大学学报. 2012，38（2）.

第 4 章　Kinect 体感交互实验

4.1　实验目的

1）了解 Kinect 体感交互系统。
2）掌握 Kinect 动作识别方法，实现在智能家居中的应用。

4.2　实验内容

1）通过创建用户的骨骼图，检测出人体的动作。
2）迅速识别出每个动作的具体含义。
3）能够完成对电视、电灯、空调等家电的控制。
4）把每个动作所代表的控制信息传送给家用电器。

4.3　实验所用仪表及设备

1）硬件：Web 服务器、数据库服务器、数据采集服务器、Kinect for Xbox 360 Sensor。
2）软件：Windows 7、Microsoft SQL Server 2008、Visual Studio 2010 Express 或者 Visual Studio 2010 专业版、. NET Framework 4. 0、KinectDeveloperToolkit- v1. 7. 0- Setup、KinectSDK- v1. 7- Setup。

4.4　实验原理

4.4.1　Kinect 设备

Kinect 是微软为其 Xbox 360 游戏主机和 Windows 平台 PC 打造的一款运动感知输入设备，如图 4-1 所示。作为一款体感外设，它实际上是一个采用全新空间定位技术（Light Coding）的 3D 体感摄像头，利用即时动态捕捉、影像辨识、麦克风输入、语音辨识、社群互动等功能，允许玩家使用身体姿势和语音命令，通过自然用户界面技术与 Xbox 360 交互，从而完全摆脱了传统游戏手柄的束缚。

Kinect 有 3 个镜头，如图 4-2 所示，中间的镜头是 RGB 彩色摄影机，左右两边的镜头分别为由红外线发射器和红外线 CMOS 摄影机所构成的 3D 深度感应器。

RGB 摄像头：为 Xbox 360 提供红、绿、蓝三种颜色通道，主要用于面部识别和动作追踪。

深度传感器：由红外线投影机和单色 CMOS 传感器组成，Xbox 360 通过计算得出空间数据。

图 4-1　Kinect 设备

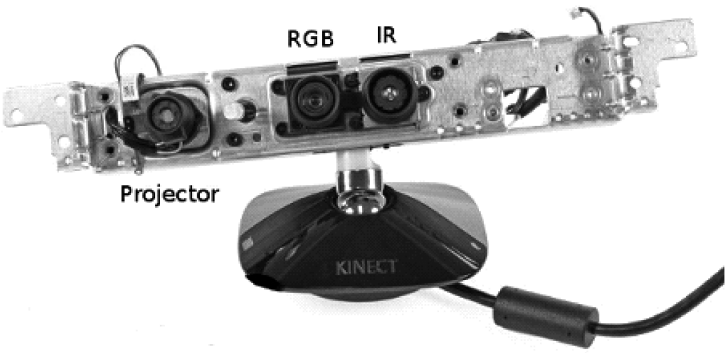


图 4-2　Kinect 内部结构

多点阵麦克风：用于聊天以及语音命令识别，以及过滤环境噪声。

4.4.2　Kinect 工作原理

Light Coding，顾名思义就是用光源照明给需要测量的空间编码，即结构光技术。与传统的结构光方法不同的是，它的光源并不是一副周期性变化的二维图像编码，而是一个具有三维纵深的“体编码”。这种光源叫做激光散斑（laser Speckle），是激光照射到粗糙物体或穿透毛玻璃后形成的随机衍射斑点。

Kinect 通过 IR 投射一些“随机”点阵，然后使用 CMOS 传感器来捕捉这些点阵。当场景的深度发生变化时，摄像头看到的点阵也会发生变化，通过这个变化可以推断出深度信息。

4.4.3　Kinect 应用领域

第一种是自然人机交互界面。用手指非接触即可操纵大屏幕上的显示内容。例如 Kinect 虚拟试衣间、Kinect 车展演示广告、Kinect 手术室影像操作，还有 Kinect 操作幻灯片、Kinect进行照片浏览等，这些自然人机交互界面的应用带来了新的用户体验。

第二种是自然环境识别，比如根据 Kinect 产生的深度数据和红外数据对物体进行三维扫描重建，利用 Kinect 进行机器人导航和障碍物自动回避等。

第三种是自然用户的识别，比如说利用 Kinect 来进行姿势识别、人脸识别。例如 Xbox360 中的舞林大会、运动会、大冒险等体感游戏，这些都是利用 Kinect 的一些数据来进行用户的动作识别，从而参与到游戏中的互动。

2011 年 12 月 2 日，由微软亚洲研究院举办的 Kinect for Windows 研讨会在北京召开，吸引了来自众多行业和研究领域的专家学者以及全国各地的大学老师和学生。研讨会就 Kinect 的体感交互技术及其应用领域进行了交流和讨论，并且展示了国内基于 Kinect for Windows SDK 在各个应用领域的研发成果。下面选择一些有代表性的项目进行简要介绍。

1. 基于 Kinect 的手语翻译系统

手语翻译系统旨在解决聋哑人与正常人的沟通问题，利用 Kinect 对肢体动作的实时捕捉，对特定的手语动作进行识别，最终翻译成文字，这样不懂手语的人也可以与聋哑人正常交流了。目前，该项目的研究已经取得了重大的进展，相信结合 Kinect 强大的体感交互技术，在不远的将来 Kinect 手语翻译系统将成为聋哑人的得力助手。

2. 空中手写

空中手写软件巧妙地利用了 Kinect 对手部节点的实时追踪，用户只需对着 Kinect 在空气中比划，便能写出相应的汉字，并输入到计算机中。该项目为未来的手写输入提供了新的思路。

3. 虚拟试衣系统

虚拟试衣系统实现了不需用户真正穿上衣服便可看到衣服上身效果的虚拟试衣体验。用户只需站在屏幕前，选择虚拟试衣系统中存储的品牌衣服图片，系统会根据 Kinect 获取到的骨架数据自适应地穿到用户在屏幕中的影像上，达到轻轻松松试衣的效果。这样不仅使得试衣变得更加便捷、有趣，而且能有效减少试衣成本。

4. Kinect 版水果忍者

Kinect 版水果忍者也是利用了 Kinect 的骨骼追踪技术，将触屏版的水果忍者游戏移植到 Kinect 的体感控制上。Kinect 追踪玩家双手的移动轨迹，玩家只需对着屏幕划动双手便可切下水果，俨然真实版忍者，在娱乐的同时还能锻炼身体。

4.4.4 Kinect 在智能家居中的应用

用户站在 Kinect 前 1 ~ 3m 的范围，做出相应动作用来控制不同的家电设备，如图 4-3 和图 4-4 所示。

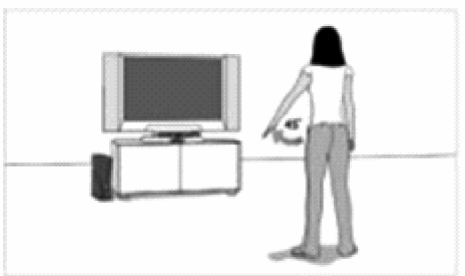


图 4-3 Kinect 的“通用暂停或退出”姿势

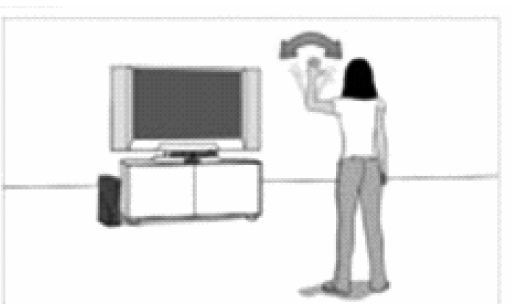


图 4-4 人挥动左手

4.5　实验步骤

4.5.1　Kinect 设备连接

1. 安装 Kinect-SDK

按提示安装，然后选择存放的路径，如图 4-5 和图 4-6 所示。

图 4-5　选择安装

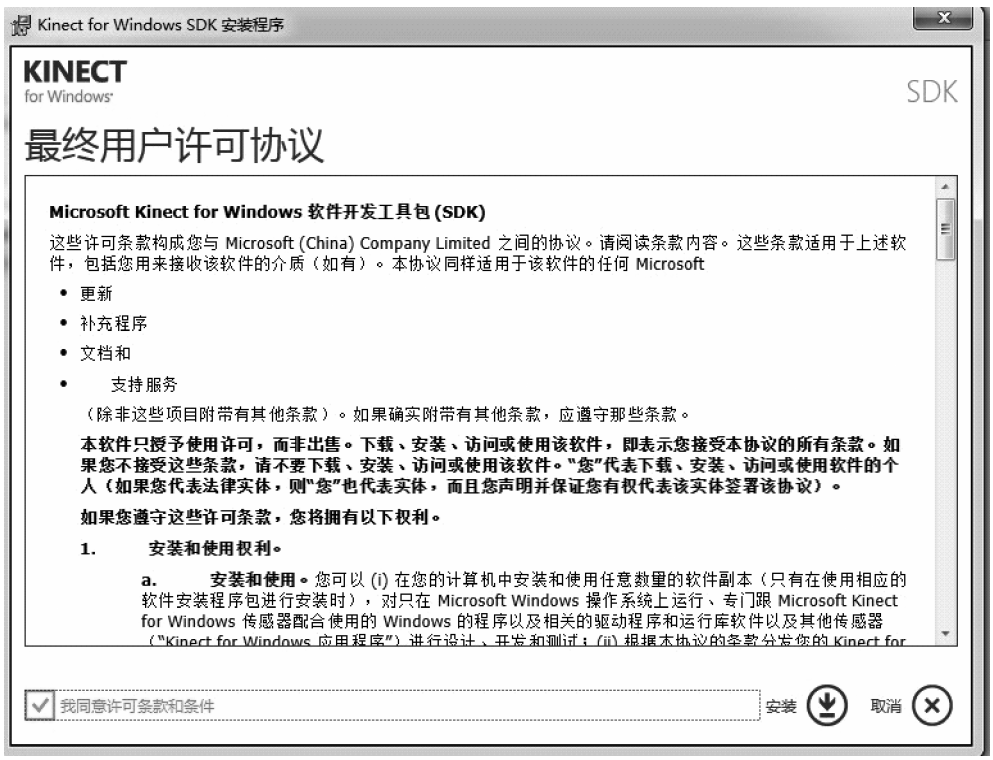


图 4-6　Kinect for Windows SDK 安装程序

2. 安装 Kinect 驱动

首先，需要断开 Kinect 与计算机的连接，并关闭 Visual Studio。等 SDK 安装完成之后，将 Kinect 电源线插上并连接到计算机上，Win7 会自动寻找和安装驱动，注意这个时候要保持计算机联网，硬件驱动的安装过程需要连接互联网。安装成功后 Kinect 的指示灯会变成绿色。

可以到计算机的设备管理器中查看。如图 4-7 所示，在 Microsoft Kinect 节点下应该有 3 个项，分别是 Microsoft Kinect Audio Array Control、Microsoft Kinect Camera 和 Microsoft Kinect

Security Control。

Kinect 麦克风是否安装成功可以在设备管理器的声音视频游戏控制器节点下查看，如图 4-8 所示，Kinect USB Audio 应该在这个节点下面。

图 4-7　查看 Kinect 安装情况

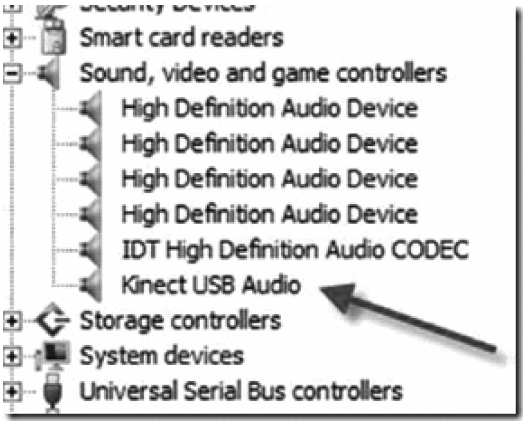


图 4-8　麦克风安装成功

4.5.2　实现过程

1. 创建工程

打开 Visual Studio 2010，单击文件→打开→项目解决方案，打开 KinectSkeletonView 项目如图 4-9 和图 4-10 所示。

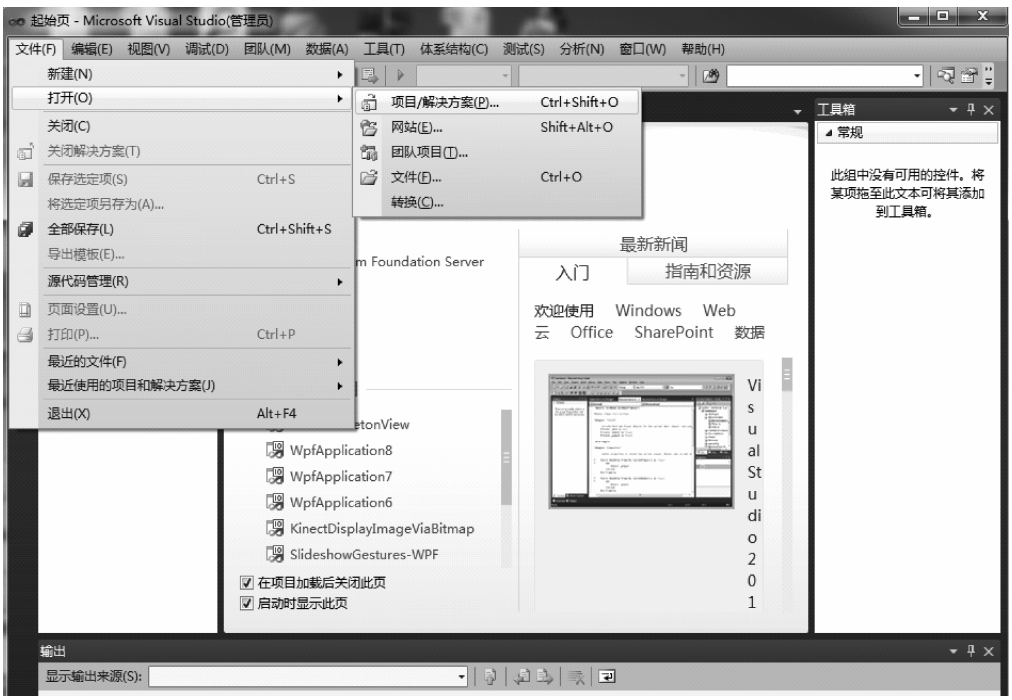


图 4-9　创建工程

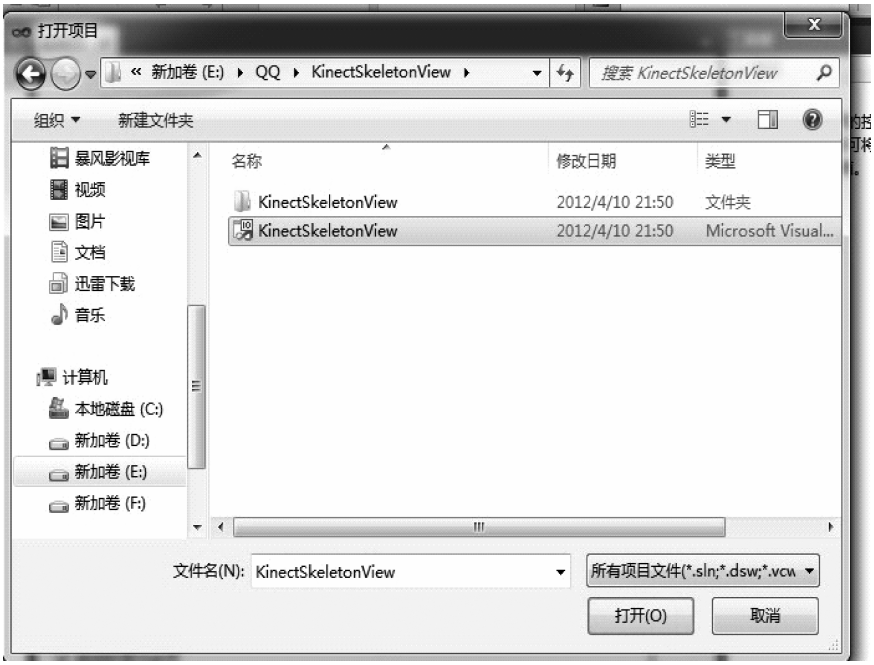


图 4-10　打开 KinectSkeletonView 项目

2. 启动调试

注意：调试人员要站在合适的距离，不要太远或者太近，一般为 1 ~ 3m。如果设备没有发现人或者发现人但是人不做任何动作，界面显示“等待动作识别……”，等待用户与Kinect有一个交互的过程如图 4-11 所示。当挥动左右手的时候能分别出现如图 4-12 和图 4-13所示的画面。

图 4-11　等待动作识别

图 4-12 挥动左手

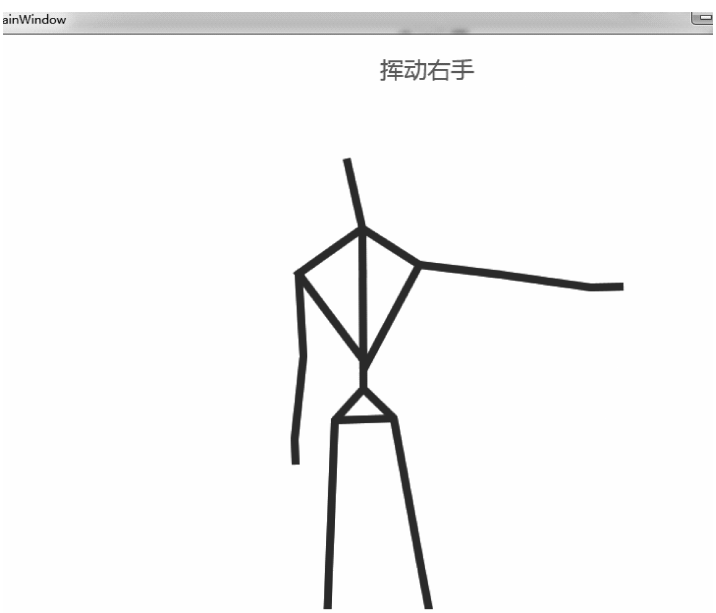


图 4-13 挥动右手

以下是该程序的代码：

```
using System;
using System. Collections. Generic;
using System. Linq;
using System. Text;
using System. Windows;
```

```
using System. Windows. Controls;
using System. Windows. Data;
using System. Windows. Documents;
using System. Windows. Input;
using System. Windows. Media;
using System. Windows. Media. Imaging;
using System. Windows. Navigation;
using System. Windows. Shapes;
using System. Net;
using Microsoft. Kinect;
namespaceKinectSkeletonView
{
/// < summary >
/// Interaction logic for MainWindow. xaml
//主函数定义一些变量
publicpartialclassMainWindow : Window
    {
privateKinectSensorkinectDevice;//定义硬件设备对象
privatereadonlyBrush[ ] skeletonBrushes;//绘图笔刷
privateSkeleton[ ] frameSkeletons;// 定义数组存储骨骼数据
publicMainWindow( )//构造函数
{
InitializeComponent( );//初始化工作
skeletonBrushes = newBrush[ ] { Brushes. Black, Brushes. Crimson, Brushes. Indigo, Brushes. Dodg-
erBlue, Brushes. Purple, Brushes. Pink};
KinectSensor. KinectSensors. StatusChanged + = KinectSensors_StatusChanged;//
this. KinectDevice = KinectSensor. KinectSensors. FirstOrDefault( x = > x. Status = = KinectStatus.
Connected);
        }
    // 检测 Kinect 前是否有人,决定是否启动设备的骨骼数据相关功能
publicKinectSensorKinectDevice
        {
get { returnthis. kinectDevice;}
set
            {
if ( this. kinectDevice! = value)                    {
//Uninitialize
if ( this. kinectDevice! = null)
                {
```

```
this. kinectDevice. Stop( ) ;
this. kinectDevice. SkeletonFrameReady- = KinectDevice_SkeletonFrameReady;
this. kinectDevice. SkeletonStream. Disable( ) ;
this. frameSkeletons = null;
                        }
this. kinectDevice = value;
//Initialize
if ( this. kinectDevice ! = null)
                        {
if ( this. kinectDevice. Status = = KinectStatus. Connected)
                            {
this. kinectDevice. SkeletonStream. Enable( ) ;
this. frameSkeletons =
newSkeleton[ this. kinectDevice. SkeletonStream. FrameSkeletonArrayLength] ;
this. kinectDevice. SkeletonFrameReady + = KinectDevice_SkeletonFrameReady;
this. kinectDevice. Start( ) ;
                            }
                        }
                    }
                }
        }
    // 判断 Kinect 的几种状态
privatevoidKinectSensors_StatusChanged( object sender, StatusChangedEventArgs e)
        {
switch ( e. Status)
            {
caseKinectStatus. Initializing:
caseKinectStatus. Connected:
caseKinectStatus. NotPowered:
caseKinectStatus. NotReady:
caseKinectStatus. DeviceNotGenuine:
this. KinectDevice = e. Sensor;
break;
caseKinectStatus. Disconnected:
//TODO: Give the user feedback to plug- in a Kinect device.
this. KinectDevice = null;
break;
default:
//TODO: Show an error state
break;
```

```
            }
        }
        //一切准备就绪时,启动骨骼数据,定义画刷将人体骨骼图画出来
privatevoidKinectDevice_SkeletonFrameReady(object sender,
SkeletonFrameReadyEventArgs e)
        {
using (SkeletonFrame frame = e. OpenSkeletonFrame())
            {
if (frame! = null)
                {
Polyline figure;
BrushuserBrush;
Skeletonskeleton;
LayoutRoot. Children. Clear();
frame. CopySkeletonDataTo(this. frameSkeletons);
Skeleton[ ]dataSet2 = newSkeleton[this. frameSkeletons. Length];
frame. CopySkeletonDataTo(dataSet2);
skeleton = (froms in dataSet2
wheres. TrackingState = = SkeletonTrackingState. Tracked
select s). FirstOrDefault();
if (skeleton! = null)
                {
ProcessGesture(skeleton);
                }
for (inti =0; i < this. frameSkeletons. Length; i + + )
                  {
skeleton = this. frameSkeletons[i];
if (skeleton. TrackingState = = SkeletonTrackingState. Tracked)
                    {
userBrush = this. skeletonBrushes[i % this. skeletonBrushes. Length];
figure = CreateFigure(skeleton,userBrush,new[ ]{JointType. Head,
JointType. ShoulderCenter,JointType. ShoulderLeft,JointType. Spine,
JointType. ShoulderRight,JointType. ShoulderCenter,JointType. HipCenter
                                                                    });
LayoutRoot. Children. Add(figure);
figure = CreateFigure(skeleton,userBrush,new[ ]{JointType. HipLeft,JointType. HipRight});
LayoutRoot. Children. Add(figure);
figure = CreateFigure(skeleton,userBrush,new[ ]{JointType. HipCenter,JointType. HipLeft,
```

```
JointType. KneeLeft,JointType. AnkleLeft,JointType. FootLeft});
LayoutRoot. Children. Add(figure);
figure = CreateFigure(skeleton,userBrush,new[]{JointType. HipCenter,JointType. HipRight,
JointType. KneeRight,JointType. AnkleRight,JointType. FootRight});
LayoutRoot. Children. Add(figure);
figure = CreateFigure(skeleton,userBrush,new[]{JointType. ShoulderLeft,JointType. ElbowLeft,
JointType. WristLeft,JointType. HandLeft});
LayoutRoot. Children. Add(figure);
figure = CreateFigure(skeleton,userBrush,new[]{JointType. ShoulderRight,JointType. ElbowRight,
JointType. WristRight,JointType. HandRight});
LayoutRoot. Children. Add(figure);
                                }
                            }
                        }
                    }
                }
    // 通过绘制一系列的直线把各个骨骼点相加连接起来组成骨骼图
privatePolylineCreateFigure(Skeleton skeleton,Brushbrush,JointType[] joints)
            {
Polyline figure = newPolyline();
figure. StrokeThickness = 8;
figure. Stroke = brush;
for (inti = 0; i < joints. Length; i + + )
                {
figure. Points. Add(GetJointPoint(skeleton. Joints[joints[i]]));
                }

return figure;

            }
    //骨骼图像的坐标在深度图像的映射
privatePointGetJointPoint(Joint joint)
            {
// JointType
DepthImagePoint point =
this. KinectDevice. CoordinateMapper. MapSkeletonPointToDepthPoint(joint. Position,
this. KinectDevice. DepthStream. Format);
point. X * = (int)this. LayoutRoot. ActualWidth/
KinectDevice. DepthStream. FrameWidth;
```

```
point. Y * = (int)this. LayoutRoot. ActualHeight/
KinectDevice. DepthStream. FrameHeight;
returnnewPoint(point. X,point. Y);
        }
int a =0;//右手
int b =0;//左手
    //  进入手势判定阶段
privatevoidProcessGesture(Skeleton s)
        {
JointleftHand = (from j ins. Jointswhere
j. JointType = =JointType. HandLeft
select j). FirstOrDefault();

JointrightHand = (from j ins. Joints
wherej. JointType = =JointType. HandRight
select j). FirstOrDefault();
Joint head = (from j ins. Joints
wherej. JointType = =JointType. Head
select j). FirstOrDefault();
if (rightHand. Position. X > head. Position. X +0. 45)
            {
                b =0;
a + +;
if(a >30)
                {
                    a =0;
                    label1. Content = "挥动右手";
//Create a request using a URL that can receive a post.
//执行开灯操作
WebRequest request =
WebRequest. Create("http://115. 25. 48. 254:8902/RestfulService/Devices/MainLamp/Call? op-
erationName = Open&param = &clientId = 123");
request. Timeout =2000;
HttpWebResponse response = (HttpWebResponse)request. GetResponse();
                }
            }
elseif (leftHand. Position. X < head. Position. X  -0. 45)
            {
                a =0;
```

```
b + +;
//执行关灯操作
if (b >30)
                    {
                        b =0;
                        label1. Content = "挥动左手…";
WebRequest request =
WebRequest. Create("http://115. 25. 48. 254:8902/RestfulService/Devices/MainLamp/Call? operationName = Close&param = &clientId = 123");
request. Timeout = 2000;
HttpWebResponse response = (HttpWebResponse) request. GetResponse();
                    }
                }
else
                {
                    label1. Content = "等待动作识别. . . . . ";
                }
            }
        }
}
```

4.6 思考题

1）软件调试中遇到的问题都有哪些？分别是怎样解决的？

2）除了本实验以外，请再想出 Kinect 与智能家居系统的其他交互方式。

4.7 实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤以及调试过程、心得体会，最后要附上程序，并按要求回答思考题 1）、2）。

参 考 文 献

[1] 倪晨，邱鹏，王峰，等. Kinect 体感技术在人机交互中的应用研究［J］. 信息技术与信息化，2013（1）.

[2] 朱涛，金国栋，芦利斌. Kinect 应用概述及发展前景［J］. 现代计算机（专业版）2013（6）.

[3] 陶丽君，李翠华，张希婧，等. 基于 Kinect 传感器深度信息的动态手势识别［J］. 厦门大学学报. 2013（7）.

[4] 谢亮，廖宏建，杨玉宝. 基于 Kinect 的姿势识别与应用研究［J］. 计算机技术与发展. 2013（5）.

[5] 林海波，梅为林，张毅，等. 基于 Kinect 骨骼信息的机械臂体感交互系统的设计［J］. 计算机应用与软件. 2013（2）.

[6] 王志良. 物联网现在与未来 [M]. 北京：机械工业出版社，2010.
[7] 王志良. 物联网工程概论 [M]. 北京：机械工业出版社，2011.
[8] 张毅，张烁，罗元，等. 基于 Kinect 深度图像信息的手势轨迹识别机应用 [J]. 计算机应用研究. 2012（9）.
[9] 郑立国，罗江林，许舸. 基于 Kinect 的动作捕捉系统的实现 [J]. 吉林大学学报. 2013（1）.
[10] 苏本跃，王广军，章健. 基于物联网环境下体感交互技术的智能家居系统 [J]. 中南大学学报. 2013（7）.

第 5 章　智能家居之 RFID 系统实验

5.1　实验目的

1）掌握 RFID 系统硬件设计、调试与焊接技术。
2）掌握 RFID 系统底层 C 语言程序编写。
3）掌握 RFID 上位机界面的设计及通信流程。
4）掌握 RFID 底层硬件与智能家居服务器端的 TCP/IP 通信。

5.2　实验内容

1）13.56MHz RFID 系统硬件的焊接与调试。
2）RFID 系统底层 C 语言驱动阅读器芯片程序的编写。
3）利用 LabWindows/CVI 编写上位机界面程序。
4）编写 VC、C#上位机程序，实现上位机与服务器端进行 Socket 通信。

5.3　实验所用仪表及设备

1）硬件：PC 一台、RFID 未焊接套件一套、USB 转串口下载线一个、电烙铁等焊接调试工具。

2）软件：Altium Designer、单片机 C 语言编程开发环境 Keil uVision4、LabWindowsCVI、STC/ISP 下载软件。

5.4　实验原理

5.4.1　RFID 概述

RFID 是 Radio Frequency Identification 的缩写，即射频识别，俗称电子标签。RFID 是一种简单的无线系统，该系统用于控制、检测和跟踪物体。RFID 是一种非接触式的自动识别技术，它通过射频信号自动识别目标对象并获取相关数据，识别工作无须人工干预，可工作于各种恶劣环境。RFID 技术可识别高速运动物体并可同时识别多个标签，操作快捷方便。

RFID 系统是由阅读器、应答器和高层等部分组成的。应答器通过射频信号向阅读器传输数据，阅读器可将高层的读/写等命令传送到应答器，也可将应答器返回的数据传送到高层，高层系统负责完成数据信息的存储、管理和控制，如图 5-1 所示。

简单的 RFID 应用系统由单个阅读器和单个应答器组成，阅读器一次仅对一个应答器进

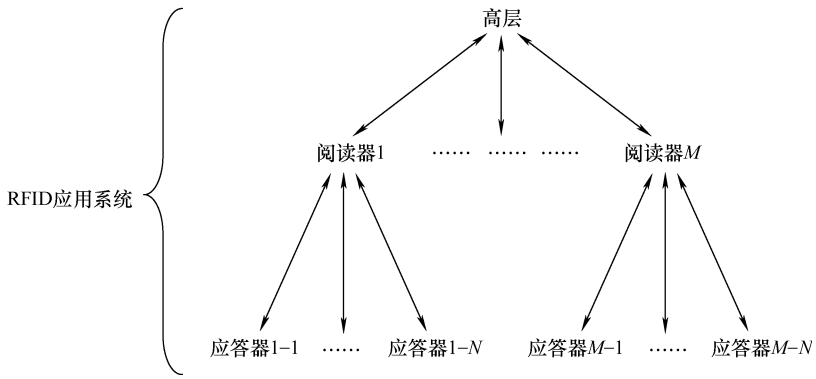


图 5-1　RFID 应用系统的结构

行操作，例如校园卡和公交卡刷卡操作。较复杂的 RFID 应用系统由单个阅读器和多个应答器组成，一个阅读器可以同时处理多个应答器，但同时需要考虑一些防冲突（防碰撞）等机制。复杂的 RFID 应用系统由高层将多个阅读器连成一个网络，其中每个阅读器仍然可以对一个或多个应答器进行操作。

1. 应答器

（1）定义

应答器，根据应用场合的不同又被称为电子标签、射频标签、射频卡和标签等。应答器是数据信息的载体，通常由耦合元件（天线）及微电子芯片组成，芯片用来存储数据，天线用来收发无线电波。

（2）性能参数

应答器的主要性能参数有工作频率、读/写能力、编码调制方式、数据传输速率、信息数据存储容量、工作距离、防冲突（防碰撞）能力和安全性能等。

（3）功能特征

应答器与阅读器之间通过电磁波进行通信，由于许多因素（如应用需求的不同，天线尺寸的不同等）而造成不同种类的应答器的功能也会有所不同。一般来说会具有以下功能：

1）具有一定的存储容量，可以存储被识别物体的相关信息；

2）可以维持被识别物品信息的完整性，并随时将信息传输给阅读器；

3）具有确定的使用年限，使用期内不需要维修；

4）对接收到的信号和需要返回给阅读器的信号进行解调、解码等各种处理；

5）收集阅读器发射到空间的电磁波，并把自身发射额定能量以电磁波的形式发射出去。

（4）分类

根据供电方式分类：无源（被动式）应答器、半无源（半被动式）应答器、有源（主动式）应答器。

根据结构形式分类，有卡形、环形、纽扣型、钥匙扣型、手表型等，不同的形状适应于不同的应用。

（5）结构（见图 5-2）

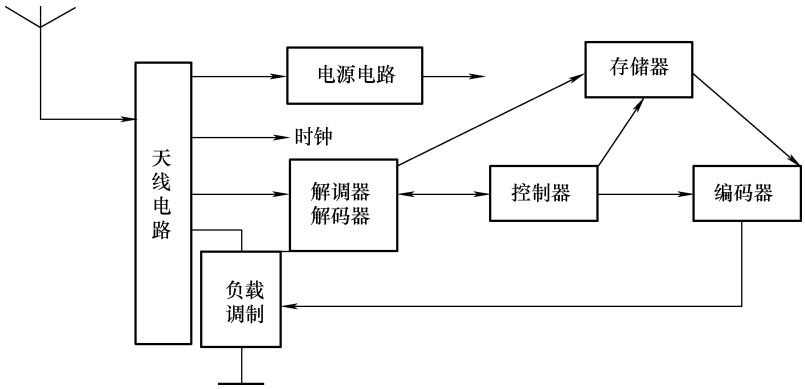


图 5-2 应答器电路的基本结构

2. 阅读器

（1）定义

阅读器又称读写器、读卡器、读头等。阅读器是 RFID 系统的信息控制和处理中心，主要负责与应答器的双向通信，同时接受来自主机的控制命令，通常由射频接口、逻辑控制单元和天线三部分组成。

（2）功能

根据频率范围、通信协议、数据传输的方法的不同，各种阅读器的作用会有一定的差异，但一般来说都具有下述功能：

1）以射频方式向应答器传输能量；

2）从应答器中读出数据或向应答器中写入数据；

3）完成对读取数据的信息处理并实现应用操作；

4）一定情况下，应能与高层处理交互信息。

（3）结构（见图 5-3）

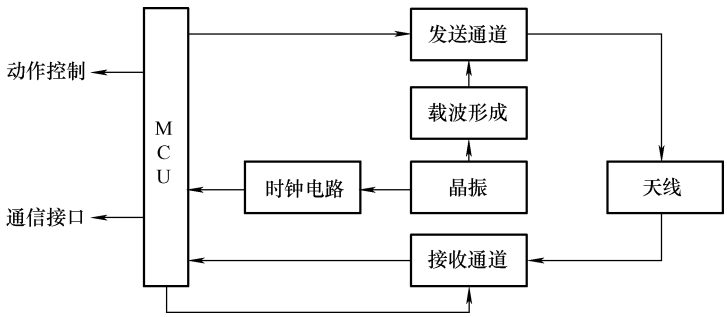


图 5-3 阅读器电路的结构框图

（4）阅读器电路设计

阅读器电路分三个模块，分别是射频模块、控制处理模块和天线模块，如图 5-4 所示。

本实验的阅读器电路的硬件设计将射频模块和天线模块一起视为天线部分，而控制处理模块视作单独的主控板部分。

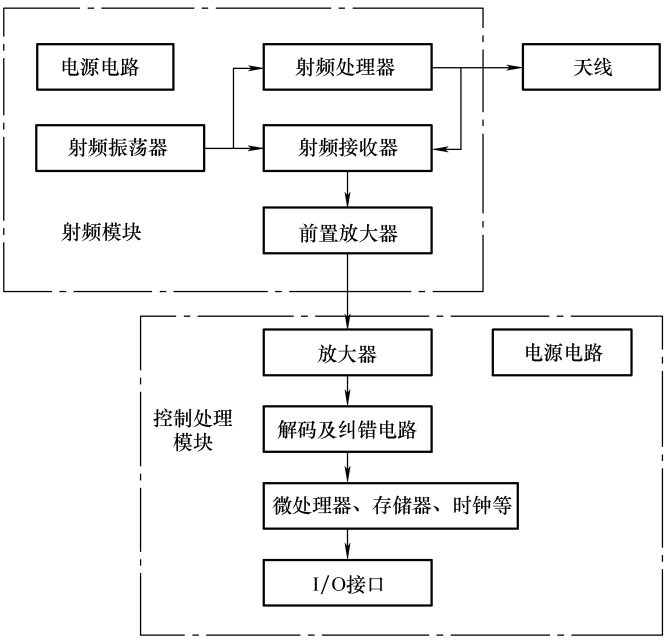


图 5-4　阅读器电路设计模块

3. 高层

对于多阅读器构成的信息系统的具体应用，需要在高层将多阅读器获取的数据进行有效的整合，以提供查询历史档案等相关管理和服务，进一步通过对数据的加工、分析和挖掘，为正确决策提供依据。

4. RFID 系统工作流程

RFID 通过无线射频方式在阅读器和应答器之间进行非接触双向数据传输，以达到目标识别和数据交换的目的。RFID 系统的一般工作流程是：

1）阅读器通过发射天线发送一定频率的射频信号；

2）当应答器进入阅读器天线工作区时，应答器天线产生足够的感应电流，应答器获得能量被激活；

3）应答器将自身信息通过内置天线发送出去；

4）阅读器天线接收到从应答器发送来的载波信号；

5）阅读器天线将载波信号传送到阅读器；

6）阅读器对接收到的信号进行解调和解码，然后送到系统高层进行相关处理；

7）系统高层根据逻辑运算判断该应答器的合法性；

8）系统高层针对不同的设定做出相应的处理，发出指令信号控制执行机构动作。

5.4.2　不同频段 RFID 产品特性及应用

不同频段的 RFID 产品（见表 5-1）会有不同的特性。目前定义 RFID 产品的工作频率有低频、高频和超高频等，不同频段有符合不同标准的各种产品，其中应答器有无源、半无源和有源几种方式。

表 5-1 不同频段 RFID 应用一览表

非接触智能卡应用 IC	城市一卡通，企业一卡通，IC 卡，校园一卡通，会员一卡通	ISO14443A	13. 56. MHz	高频	距离 10cm，通信速率高，适合电子钱包应用，多数加密钱包基于此
	手机支付	ISO18092/NFC IP-1			空中协议 14443A
	中国二代身份证	ISO14443B			
	中国香港八达通卡	FELICA			
物品识别应用 RFID	畜牧业，工业现场，企业一卡通 ID 卡	ISO11784/S ISO14223	125～134kHz	低频	穿透水，金属相对不敏感；通信速率低，不适合高速读取，距离 1～2m
	数字图书馆，洗衣，生产线，人员通道	ISO15693/18000-3 HF-EPC NFC IP-2	13. 65MHz	高频	距离最远可达 1m 多，是高频的物流标准，透水性好，边界清晰，区域定位好
	车辆管理，资产管理，物流管理	ISO 18000-6B/C UHF-EPC	840～845MHz 920～925MHz	超高频	标签价格低，通信速度快，距离远，固定阅读器可达 10m 以上，手持机也可达 3m 以上，适合商品物流，车辆管理的需要；透水性差，区域定位差
	WSN（无线传感网络）	ZigBee，802. 15. 4	2. 45GHz	微波/有源	功率一般在 10mW 数量级，距离可以达到 100～1000m
	RTLS，人员定位	无线以太网 Wi-Fi	2. 45GHz	微波/有源	
	国家高速公路不停车收费	ETC	5. 8GHz	微波/有源	
	车辆管理等，用于停车场	蓝牙	2. 45GHz	微波/有源	皆基于蓝牙的私有协议，无业界标准
	车辆管理	私有协议	2. 45GHz	微波/有源	功率一般在 mW 数量级，距离可以达到 100m 以内
	轮船等行业	ISO 18000-7	433MHz	超高频/有源	有源的距离可以从 10m 到上百米

5. 4. 3 RFID 应用

RFID 的应用领域极其广阔，并已在欧美等发达国家得以大量成功应用，例如，航空行李的识别管理、邮政快速包裹的分拣发送、铁路列车的高速识别、钢铁厂原料统计、垃圾处理场垃圾车计量、水泥预制件厂搅拌车出入管理、图书馆图书标签、物资仓储管理以及有价证券、重要证件、高档商品的防伪处理，财产、公文管理，甚至植入动物体内用于科学跟踪研究等。为此，国外不少厂商纷纷涉足此市场颇丰、前景看好的相关产品的研制，如 Motorola、Texas、Philips、Temic、Tagmaster、Melexis、Microelectronic-Marin 等。在我国，RFID 已应用到各个领域，如餐厅刷卡系统、公交车收费系统、门禁系统、图书馆管理、购物消费以及生产线自动化。各种产品如图 5-5 所示。

图 5-5　现实生活中的各种 RFID 系统产品

5.4.4　本实验所用的 RFID 系统

1. RFID 系统组成

如图 5-6 所示，典型 RFID 系统由 RFID 阅读器、天线、应答器（射频卡）构成。阅读器通过天线发送一定频率的射频信号，当应答器（射频卡）进入磁场时产生感应电流从而获得能量，发出自身编码等信息，被阅读器读取并解码送至主控芯片及 PC 可根据命令进行操作处理。

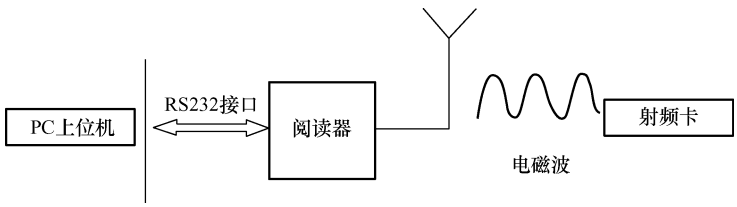


图 5-6　射频识别系统工作原理图

2. RFID 系统总体结构

要实现 RFID 系统功能要求，系统的硬件方案考虑如下，主控部分对音频、网络、数据复杂处理都没有涉及，不用上操作系统，采用 8 位机即可完成相应功能，本系统可采用基于 51 的 8 位机即可。上位机用 C#、VC + + 、LabWindows/CVI 均能实现。天线电路板要采用专用的读卡芯片，射频卡采用的是非接触式 IC 薄卡 Mifare，本系统采用的是 Type A 型卡。

根据系统功能要求可得到如图 5-7 所示的系统结构图。

3. 技术指标

（1）读卡芯片及射频卡技术指标

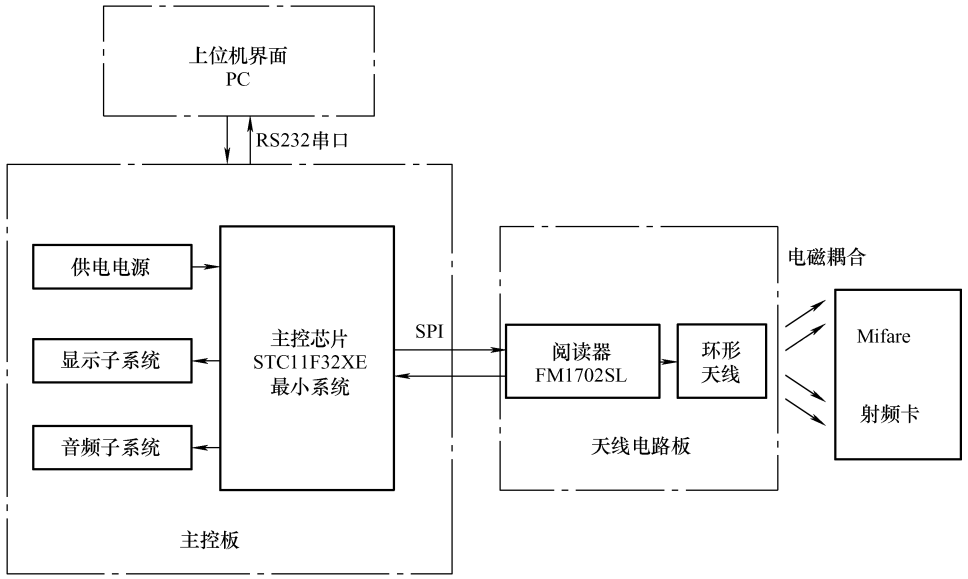


图 5-7 RFID 系统结构图技术指标

工作频率：13.56MHz。

支持射频卡类型：非接触；无源；Mifare 卡；Type A 型卡。

（2）阅读器详细技术指标

供电电压：5V。

产品型号：RFID-FM1702SL。

工作电流：13～26mA。

空闲电流：10～13mA

休眠电流：＜80μA。

峰值电流：＜30mA。

工作频率：13.56MHz。

读卡距离：（3～5）cm（Mifare 卡）。

接口：SPI。

数据传输速率：最大 10Mbit/s。

支持卡类型：Mifare S50、Mifare S70、Mifare UltraLight、Mifare Pro、Mifare Desfire。

物理特性：尺寸：40mm×40mm。

工作温度：－20～80℃。

存储温度：－40～85℃。

湿度：相对湿度 5%～95%。

4. 详细设计

（1）硬件设计与元器件选择

RFID 系统硬件结构由三大部分构成，分别为主控单元、阅读器和天线电路板、射频卡三部分。

主控部分由 51 单片机作为控制核心，采用的是 STC11F32XE，此芯片采用的是 51 内核

的单片机，硬件和指令兼容 51 单片机，主控板由 6 位一体数码管作为显示，蜂鸣器为音频模块。

阅读器和天线电路板采用复旦微电子股份有限公司的 FM1702SL 的基于 ISO14443 标准的非接触卡读卡机专用芯片，支持 13.56MHz 频率下的 TypeA 非接触通信协议，支持多种加密算法，可外置环形天线与（射频卡）进行数据通信。

主控板与天线电路板的通信采用的是 SPI 总线的方式，由于 STC11F32XE 单片机内部没有 SPI 模块，所以采用的是用单片机的 I/O 口进行模拟 SPI 总线以实现与阅读器芯片 FM1702SL 的 SPI 进行数据通信。

射频卡采用的是非接触式 IC 薄卡 Mifare 1K（S50），它是由 IC 芯片及感应天线组成的，经高温层压工艺压制成国际标准尺寸的卡片。Mifare S50 射频薄卡采用的飞利浦（NXP）原装的 Mifare IC S50 芯片，符合 IEC/ISO 14443A 空气接口协议。其具有先进的数据加密及双向密码验证系统和 16 个完全独立的扇区，有着极高的稳定性和广泛的应用范围，是企业一卡通、水表预付费、公交储值卡、高速公路收费、停车场、小区管理、交运卡、公园和公路等首选的 RFID 产品，该卡属于 TypeA 型卡。

（2）硬件原理图设计

根据以上要求，可用 Altium Designer 6.9 软件完成原理图的绘制，主控板电路分电源模块、数码管显示子系统、MAX232 电平转换模块、蜂鸣器模块、天线电路接口部分。天线电路板采用的复旦微电子的 FM1702SL 专用读卡芯片，只需要根据芯片手册设计出相应外围电路，其中天线部分是设计的关键，采用的是在电路板上绘制的环形天线。

主控板原理图如图 5-8 所示。

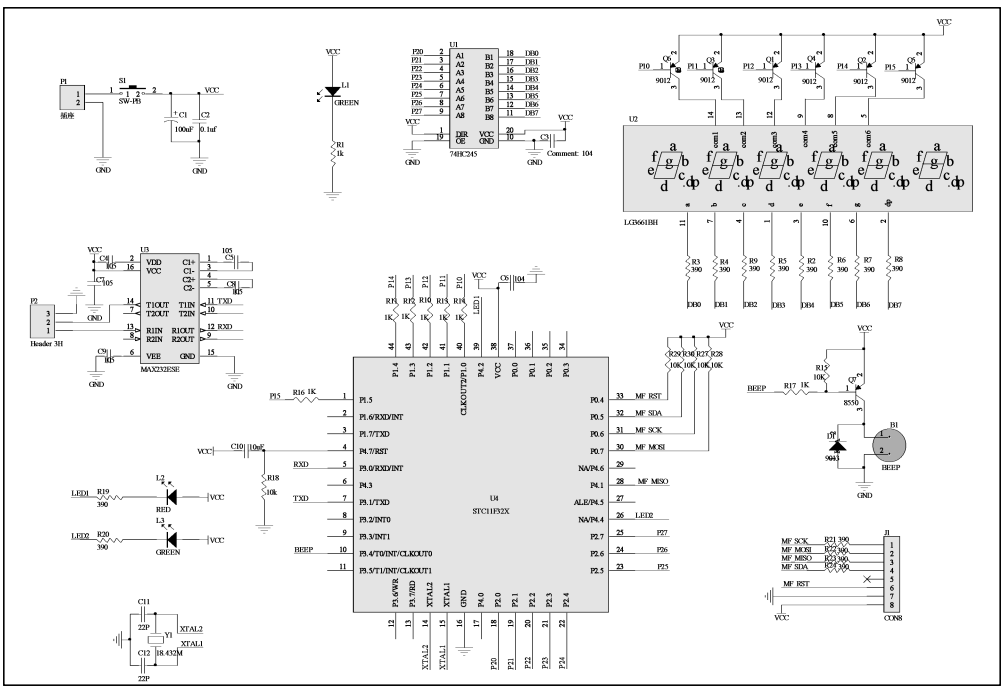


图 5-8　主控板原理图

天线电路板原理图如图 5-9 所示。

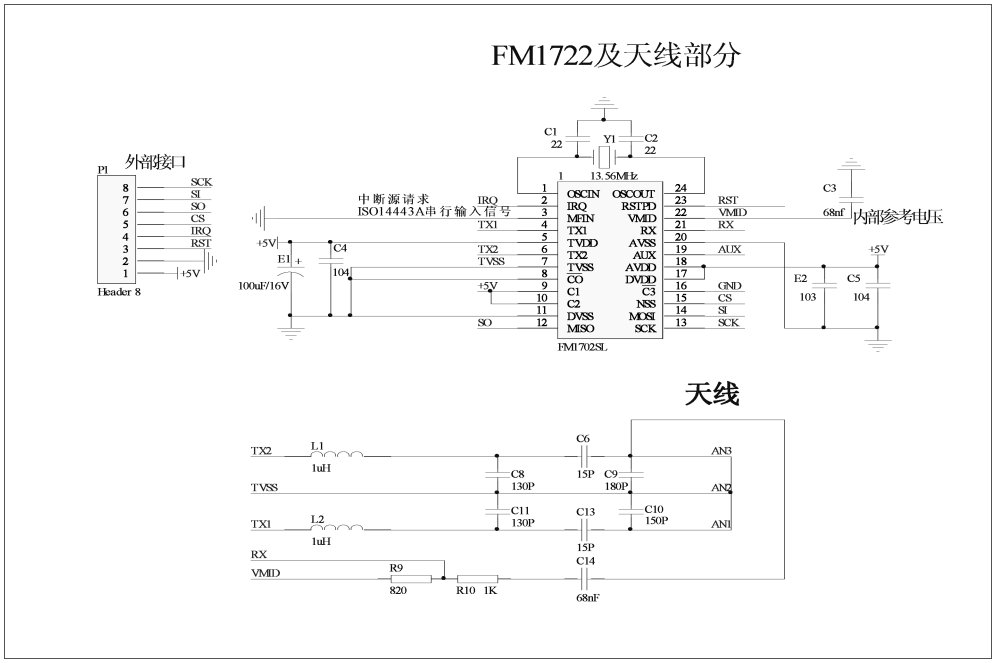


图 5-9 天线电路板原理图

（3）PCB 设计及成本估算

主控板 PCB 设计如图 5-10 所示，用的是 10cm × 10cm 的双层板大小，绘图软件为 Altium Designer 6.9。

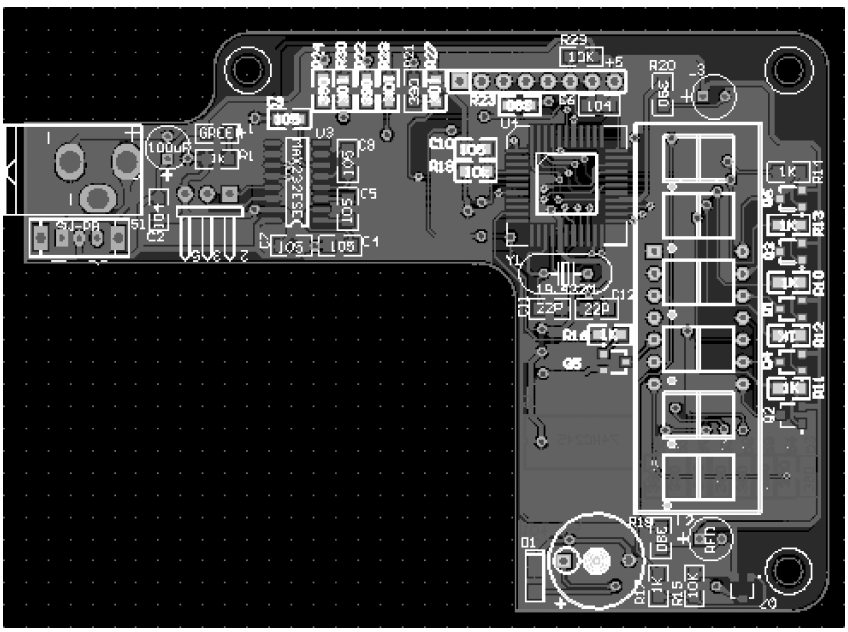


图 5-10 主控板 PCB

天线电路板 PCB 设计如图 5-11 所示。

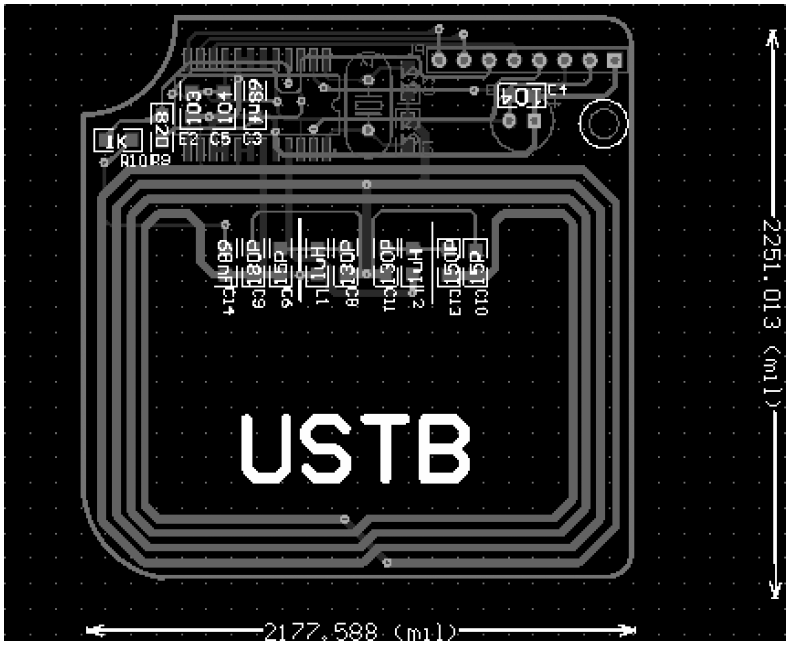


图 5-11　天线电路板 PCB

系统设计成本估算：根据以上设计，可得到系统所用的元器件及费用估算。

主控板 10cm × 10cm 大致 10 片共约 200 元，天线电路板稍小一点大致 10 片 150 元，所以 PCB 部分一套设计下来大致为 35 元。

元器件费用：STC11F32XE 淘宝网上卖大致为 5 元，天线电路板上阅读器 FM1702SL 为 10 元左右，MAX232 芯片 2 元，74HC245 驱动芯片 1 元，6 位一体数码管 6 元，其他的很多电阻电容价钱都很低了，所有加一起共 10 元左右。如果要带上外壳，还要另外再购买，整个系统设计完大致 70 ~ 100 元。

（4）软件开发环境

单片机底层硬件 C 语言编程采用的是 Keil uVision4 开发软件，上位机采用 LabWindows / CVI 作为开发环境。

LabWindows/CVI 是 National Instruments 公司（美国国家仪器公司，简称 NI 公司）推出的交互式 C 语言开发平台。LabWindows/CVI 将功能强大、使用灵活的 C 语言平台与用于数据采集分析和显示的测控专业工具有机地结合起来，利用它的集成化开发环境、交互式编程方法、函数面板和丰富的库函数大大增强了 C 语言的功能，为熟悉 C 语言的开发设计人员编写检测系统、自动测试环境、数据采集系统、过程监控系统等应用软件提供了一个理想的软件开发环境。

上位机完成界面部分，负责完成与底层的通信交互。本次设计中完成了一个消费刷卡的界面，如图 5-12 所示。

（5）不干胶电子标签

在本实验的后期，使用了不干胶电子标签。与之前用的射频白卡相比，不干胶电子标签

图 5-12 上位机操作界面

体积更小，携带更方便，应用范围相当广泛。如图 5-13 所示。

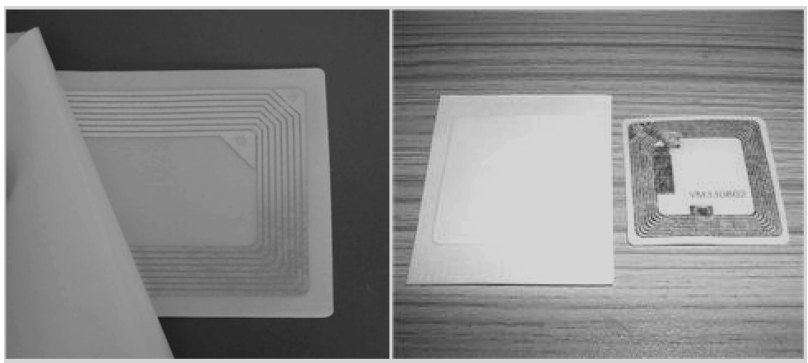


图 5-13 不干胶电子标签

把标签的天线嵌进塑料物体当中，通常采用蚀刻、电镀技术；嵌入到纸张当中，则采用胶版印刷、丝网印刷和胶印技术，仅通过一个简单的热蒸发过程就可以让铝质线圈天线在纸上成形，并组成完整的 RFID 标签。铝比铜或银更便宜，目前已经在各种标签中广泛使用。研究表明，该方法会降低 1/5 的成本。

不干胶电子标签用纸、合成纸或 PET 作为基层材料，采用银天线印刷工艺制作，单层结构，不需制作 Inlay，适用于图书馆、物流、航空行李标签、邮政信函、安全管理等射频识别（RFID）应用领域。

1）尺寸选择（见图 5-14）。

2）标签简况。

天线材料：铝；

基材材料：PET；

天线厚度：0.03mm；

产品编号	CMMF1	CMMF2	CMMF3	CMMF4	CMMF5	CMMF6	CMMF7	CMMF8	CMMF9
标签尺寸/mm	86×54	73×23	55×17	50×50	40×25	35×35	30×15	ϕ45	ϕ25
天线尺寸/mm	75×45	52×15	52×15	45×45	38×22	32×32	23×11	ϕ42	ϕ22

图 5-14　不干胶电子标签尺寸选择

基材厚度：0.03mm；

感应频率：14.0 ±0.5MHz；

读写距离：0 ~10CM；

符合 ISO14443A 标准。

3）芯片参数。

芯片可选：NXP S50/S70/UL/FM1108/Desfire 等；

工作频率：13.56MHz；

谐振频率：14.0MHz ±0.5MHz；

典型激活场强（读）：94dBμA/m#；

典型激活场强（写）：97dBμA/m#；

可擦写次数：10 万次；

数据保存时间：10 年；

通信速率：106 k bits/s。

4）环境参数。

理想存储环境：15 ~25℃，20% RH ~45% RH；

容忍存储环境：-25 ~50℃，20% RH ~90% RH；

容忍工作环境：-25 ~65℃，20% RH ~90% RH；

耐弯能力：≥ϕ20mm。

5）应用场合。

不干胶电子标签广泛应用于货物供应管理、包裹和邮政服务，物流仓储管理系统、图书馆和出租服务、航空行李箱标签、产品证明、品牌防伪保护、通常的识别系统、压容器标签等非金属物品物流贴纸。

（6）NFC

1）NFC 简介。NFC 即 Near Field Communication，译为近距离无线通信技术，亦即近场通信。NFC 是由飞利浦公司、诺基亚公司和索尼公司发起的一种非接触式识别和互联技术，是非接触 IC 卡应用融合移动终端的主流技术。

NFC 是一种短距离的高频无线通信技术，允许电子设备之间进行非接触式点对点数据传输（在 10cm 内）交换数据。这个技术由非接触式射频识别（RFID）演变而来，并向下兼容 RFID，最早由 Sony 和 Philips 公司各自开发成功，主要用于手机等手持设备中提供 M2M（Machine to Machine）的通信。由于近场通信具有天然的安全性，因此，NFC 技术被

认为在手机支付等领域具有很大的应用前景。同时，NFC 也因为其相比于其他无线通信技术较好的安全性被中国物联网校企联盟比作机器之间的“安全对话”。

2）NFC 与 RFID 的联系与区别。

RFID 和 NFC 都是标签技术（Tagging），NFC 是 RFID 技术的一个扩展。

NFC 是基于 13.56MHz 的改良 RFID 技术，主要是应用于移动支付和移动识读。NFC 是近年来在手机上应用较多的技术，主要是两个 NFC 设备之间，在相当近的距离内靠近时，互相交互信息，并完成对应的交易等功能。NFC 最大的特点是将阅读器和应答器整合到同一个芯片中。

RFID 可以在数米的距离内收发信息而 NFC 仅允许 10cm 的安全距离；RFID 的应用范围很大，而 NFC 仅在需要安全的情况下使用；很多手机已经装备了 NFC 设备。

NFC 与 RFID 两者的关系是，可以认为 NFC 是 RFID 的一个子集，即通信距离在 10cm 或 4inch 以内的 RFID 是 NFC。这种区别主要是由于，在一些设计现金支付、信用卡的应用中，RFID 的通信距离情况下，其他设备也可以收到个人 RFID 信息，存在不安全因素；这就是 NFC 提出的原因。如果使用一些特殊的信号采集设备，可以在相当远的距离外读取到 RFID 的信息。

NFC 与 RFID 一样，NFC 信息也是通过频谱中无线频率部分的电磁感应耦合方式传递，但两者之间还是存在很大的区别。

首先，NFC 是一种提供轻松、安全、迅速的通信无线连接技术，其传输范围比 RFID 小，RFID 的传输范围可以达到几米、甚至几十米，但由于 NFC 采取了独特的信号衰减技术，相对于 RFID 来说 NFC 具有距离近、带宽高、能耗低等特点。

其次，NFC 与现有非接触智能卡技术兼容，目前已经成为得到越来越多主要厂商支持的正式标准。

再次，NFC 还是一种近距离连接协议，提供各种设备间轻松、安全、迅速而自动的通信。与无线世界中的其他连接方式相比，NFC 是一种近距离的私密通信方式。

最后，RFID 更多地被应用在生产、物流、跟踪、资产管理上，而 NFC 则在门禁、公交、手机支付等领域发挥着巨大的作用。

有源 RFID 是一种长作用距离的射频识别技术，在停车场和高速公路不停车收费上可以有很好的应用，实现方式是通过有源 RFID 电子标签和固定/手持读写设备的信息交换。NFC 一般在数厘米的作用距离于 13.56MHz 的频段运作进行相互的数据交换，有源 RFID 在作用距离和工作频段与 NFC 都有较大的差异。NFC 目前来看更多的是针对于消费类电子设备相互通信，有源 RFID 则更擅长于长距离识别。

3）NFC 的通信模式解析和技术特点。

支持 NFC 的设备可以在主动或被动模式下交换数据。

在被动模式下，启动 NFC 通信的设备，也称为 NFC 发起设备（主设备），在整个通信过程中提供射频场（RF-field）。它可以选择 106kbit/s、212kbit/s 或 424kbit/s 其中一种传输速度，将数据发送到另一台设备。另一台设备称为 NFC 目标设备（从设备），不必产生射频场，而使用负载调制（load modulation）技术，即可以相同的速度将数据传回发起设备。此通信机制与基于 ISO14443A、Mifare 和 FeliCa 的非接触式智能卡兼容，因此，NFC 发起设备在被动模式下，可以用相同的连接和初始化过程检测非接触式智能卡或 NFC 目标设备，并

与之建立联系。

在主动模式下，每台设备要向另一台设备发送数据时，都必须产生自己的射频场。如图 5-15 所示，发起设备和目标设备都要产生自己的射频场，以便进行通信。这是对等网络通信的标准模式，可以获得非常快速的连接设置。

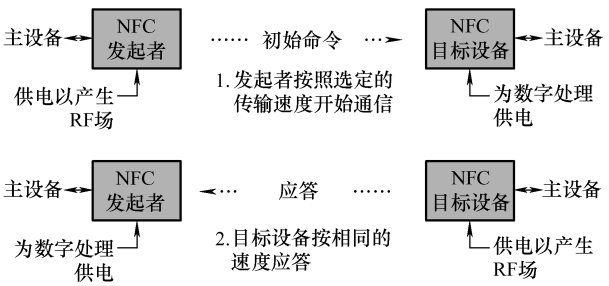


图 5-15　NFC 主动通信模式

如图 5-16 所示，移动设备主要以被动模式操作，可以大幅降低功耗，并延长电池寿命。在一个应用会话过程中，NFC 设备可以在发起设备和目标设备之间切换自己的角色。利用这项功能，电池电量较低的设备可以要求以被动模式充当目标设备，而不是发起设备。

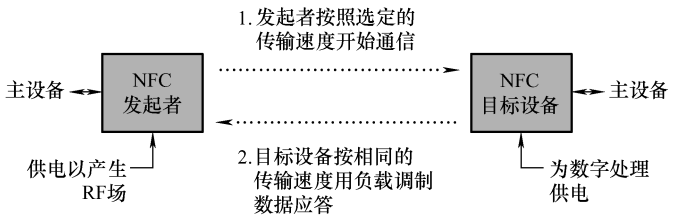


图 5-16　NFC 被动通信模式

NFC 手机内置 NFC 芯片，组成 RFID 模块的一部分，可以当做 RFID 无源标签使用——用来支付费用；也可以当做 RFID 阅读器——用做数据交换与采集。NFC 技术支持多种应用，包括移动支付与交易、对等式通信及移动中信息访问等。通过 NFC 手机，人们可以在任何地点、任何时间，通过任何设备，与他们希望得到的娱乐服务与交易联系在一起，从而完成付款，获取海报信息等。NFC 设备可以用做非接触式智能卡、智能卡的阅读器终端以及设备对设备的数据传输链路，可用于付款和购票、用于电子票证、用于智能媒体以及用于交换、传输数据等。

5.5　实验步骤

5.5.1　焊接、调试系统硬件

焊接主控板、天线电路板分别如图 5-10 和图 5-11 所示。

5.5.2　编写 C 语言程序

1. 通信协议代码

在单片机通信系统（含与 PC 上位机之间的通信）中，通信协议的设计与实现都是整个通信系统的难点和关键所在，通信协议完成了数据发送方式的规定，包含数据格式的定义和数据位的定义等。

一般通信协议要完成信号的发送与接收、差错控制以及透明性和顺序控制、链路控制与管理、对话控制、流量控制等。

本系统上位机与底层单片机采用的通信协议为：起始位（STA）+命令（CMD）+长度（LEN）+数据（DATA）+校验位（CHECK）+结束位（STOP）；其中命令位由自己设定，校验位是除本身外所有数据相加和取低 8 位，代码分析如下：

```
int   protocol(int *rev)
{
  static int process=0;                    //接收步骤标志
  unsigned char   len;                     //串口数据长度
  int length,i;
  unsigned char check=0,rev_Check;         //校验位 check、rev_Check 比对得出校验结果
  if(*rev==0x68){                          //第一位为 0x68
    process=1;                             //切换到第一步
    check+=0x68;                           //加入校验位
    *rev++;
    uart.rfid.cmd=*rev++;                  //接收第二个数据
    len=*rev++;                            //接收第三个数据
    check+=uart.rfid.cmd;                  //加入校验位
    check+=len;                            //加入校验位
    Fmt(&length,"%i<%x",len);
    if(uart.rfid.cmd==0x08){
        for(i=0;i<length;i++){
        check+=*rev;
        uart.rfid.UID[i]=*rev++;           //如果命令为 0x08 记录卡号
      }
    }
    check+=*rev;
    rev_Check=*rev++;                      //接收默认校验位
    if(*rev==0x16&&rev_Check==check){
        process=0;                         //校验成功
        return 0;
    }
    return 1;
  }
  if(process==0)
    *rev++;
  return 1;
}
```

2. FM1702SL 控制细则（见图 5-17）

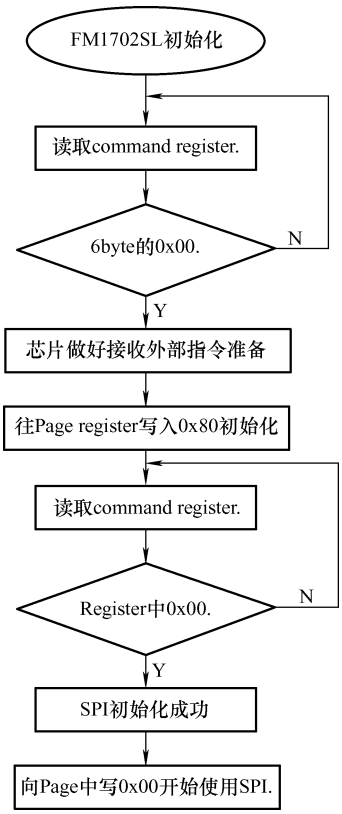


图 5-17　FM1702SL 控制细则

3. RFID 充值消费卡实例开发

RFID 是一种非接触式的自动识别技术，它通过射频信号自动识别目标对象并获取相关数据，识别工作无须人工干预，可工作于各种恶劣环境。RFID 技术具备防水、防磁、耐高温、使用寿命长、读取距离大、标签上数据可以加密、存储数据容量更大、存储信息更改自如等优点。RFID 在消费上极大地方便了我们的生活，被应用在校园卡、购物中心、餐饮娱乐等场合。

如图 5-18 所示，本例采用 RFID 套装，直接对卡进行模拟消费、充值、修改密码、查

图 5-18　RFID 套装

看余额等信息，并对 RFID 的数据进行解析，从而更好地了解卡的应用方法。

1）打开上位机软件，打开串口，如图 5-19 所示。

图 5-19　上位机软件

2）将白卡放在阅读器上，信息区显示当前卡号，如图 5-20 所示。单击“余额”，查看当前余额，同时在阅读器上也可以读出当前余额，如图 5-21 所示。

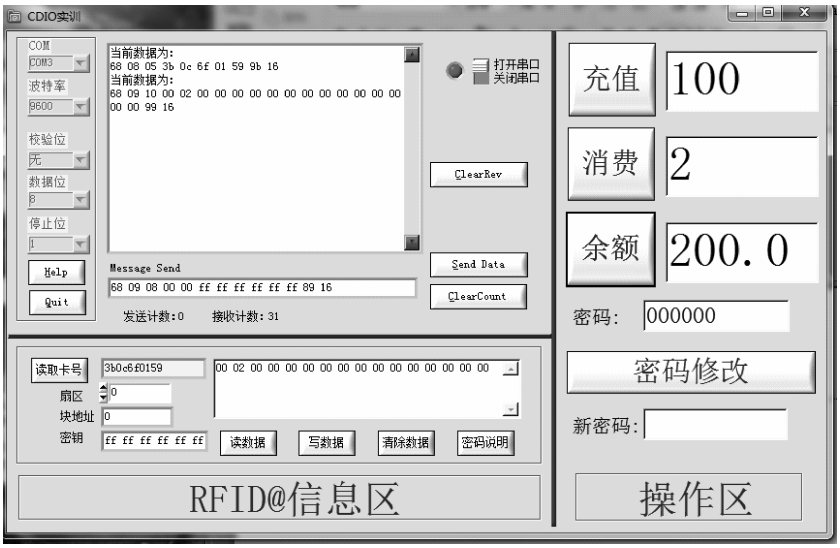


图 5-20　读取卡号

3）单击“充值”，信息区返回当前数据为：68 01 00 7f 16 表示充值成功，单击“余额”显示为 300，如图 5-22 所示；此时拿下射频卡，再放置在阅读器上显示当前余额 300，如图 5-23 所示。

4）单击“消费”，信息区返回当前数据为：68 01 00 7f 16 表示消费成功，如图 5-24 所示；单击“余额”显示为 298.0，如图 5-25 所示；此时拿下射频卡，再放置在阅读器上显

图 5-21　显示余额

图 5-22　充值操作

示当前余额 298.0，如图 5-26 所示。

5）键入新密码，单击“密码修改”，信息区返回当前数据为：68 01 00 7f 16 表示修改

图 5-23 显示充值余额

图 5-24 消费操作信息

密码成功。如果此时想操作卡，必须在密码处键入已经修改好的密码否则不会读任何数据，如图 5-27 所示。

4. 考勤系统软件开发

（1）考勤状态

针对 RFID 开发的班级考勤管理系统如图 5-28 所示，为 13.56MHz RFID 射频阅读器提供实例应用，只需要校园卡便可实现考勤。

首先实现 13.56MHz RFID 射频阅读器与软件的连接，单击系统配置→串口配置，弹出串口配置界面如图 5-29 所示，选择对应的串口，单击打开，此时考勤指示灯变成蓝色，如图 5-30 所示，进入考勤状态。

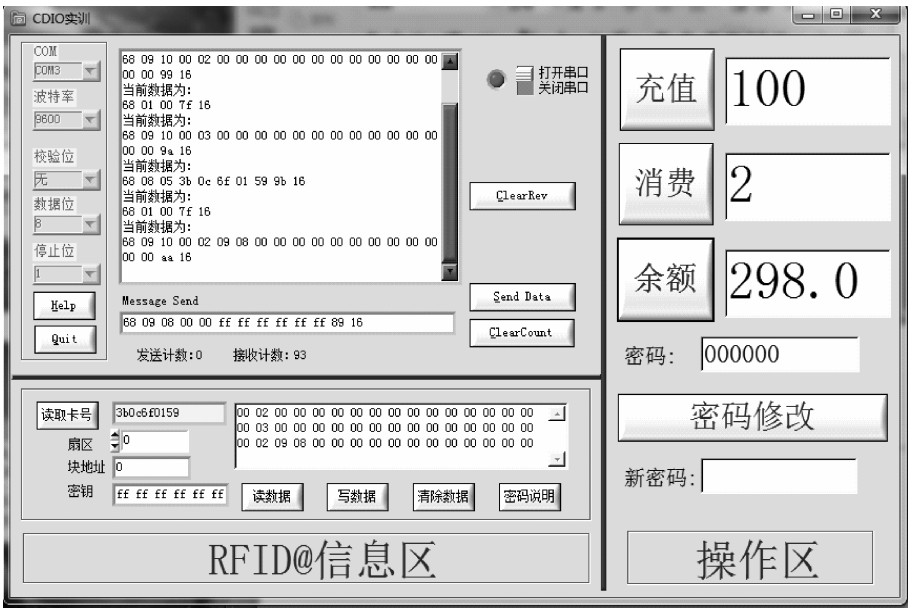


图 5-25　消费操作显示

图 5-26　显示消费余额

图 5-27　密码操作

（2）查看已经注册用户的考勤状态

单击数据信息→打开考勤表（或望远镜图标），弹出如图 5-31 所示的查找界面，输入对应姓名，单击“Find”，弹出个人考勤表（包含当前人所有的记录）如图 5-32 所示。

（3）权限设置

为了保证数据的安全，软件设有权限管理，分为两级用户：一级和二级。一级用户需要

图 5-28 软件的界面显示

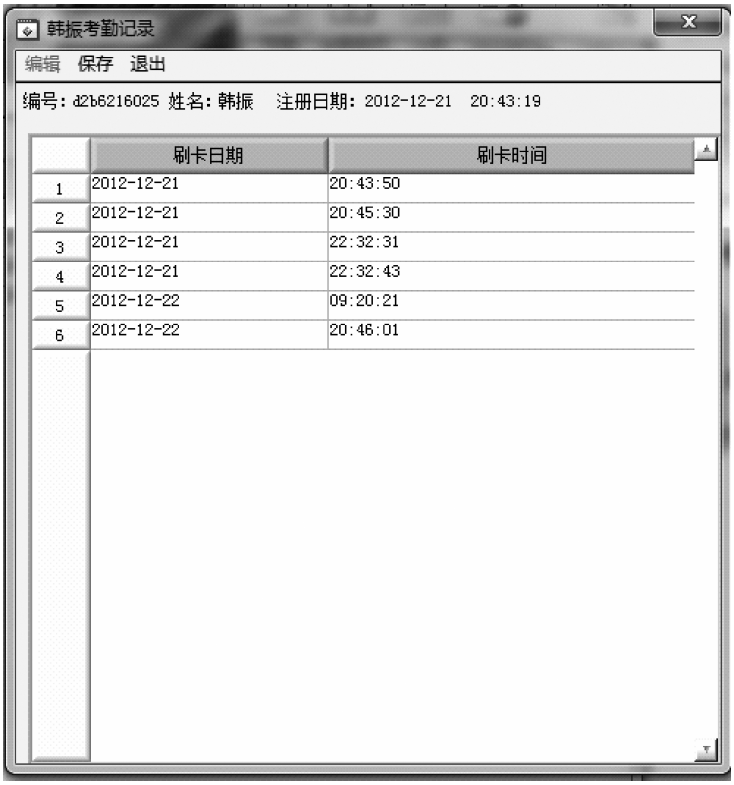
图 5-29 串口配置

通过用户登录，可进行考勤、查询、数据的修改、密钥的修改、用户的添加与删除；二级用户可进行考勤、查询，不能进行写操作。

	编号	姓名	注册日期	刷卡次数	最近刷卡时间
1	d2b6216025	韩振	2012-12-21 20:43:19	7	2013-3-12 08:23:11
2	b237388c31	陈泽平	2012-12-21 20:45:24	1	2012-12-21 20:45:33
3	b261368c69	王灵国	2012-12-21 22:32:22	2	2012-12-21 22:32:49
4	3b0c6f0159	王志良	2013-1-24 10:31:33	0	2013-1-24 10:31:33
5	6e60b7c67f	李四	2013-3-12 08:23:04	1	2013-3-12 08:23:15

图 5-30　进入考勤状态

图 5-31　查找界面



	刷卡日期	刷卡时间
1	2012-12-21	20:43:50
2	2012-12-21	20:45:30
3	2012-12-21	22:32:31
4	2012-12-21	22:32:43
5	2012-12-22	09:20:21
6	2012-12-22	20:46:01

图 5-32　个人考勤表

图 5-28 中 为密钥图标，点击可进入登录窗口，如图 5-33 所示。

（4）用户注册

用户注册是一个很重要的环节，是考勤的基础，单击用户→用户注册，弹出用户注册窗口，如图 5-34 所示，将 IC 卡置于 13.56MHz RFID 射频阅读器上，卡号自动发送至软件，输入用户名，单击“信息注册”完成注册，注册时间会显示在主窗体上，如图 5-35 所示。

图 5-33 登录窗口

图 5-34 用户注册窗口

图 5-35 注册信息

放入第二张要注册的卡片，卡号自动发送到软件，同样输入姓名，单击“信息注册”，完成第二张卡的注册即多张卡注册，如图 5-36 所示。

图 5-36　多张卡注册

5. RFID 系统与智能家居的 TCP/IP 通信

本 RFID 系统可与智能家居有良好的交互，本实验部分实现了在 RFID 设备端刷卡可以实现智能家居服务器的识别与响应，系统软件可根据卡号进行相关处理。

把 RFID 设备连接至电脑，双击 TCP1. exe 图标，打开服务器端软件，客户端界面显示如图 5-37 所示。

图 5-37　登录客户端

设置服务器端 IP 地址：115. 25. 48. 254，端口号为 1500，单击“连接”，此时客户端软件就会向服务器发起连接请求，连接成功后显示如图 5-38 所示。

图 5-38　连接服务器成功

此时把卡放在读卡区范围，RFID 设备会读到卡，并将其卡号送到客户端软件并把卡号数据发给服务器，如图 5-39 所示。

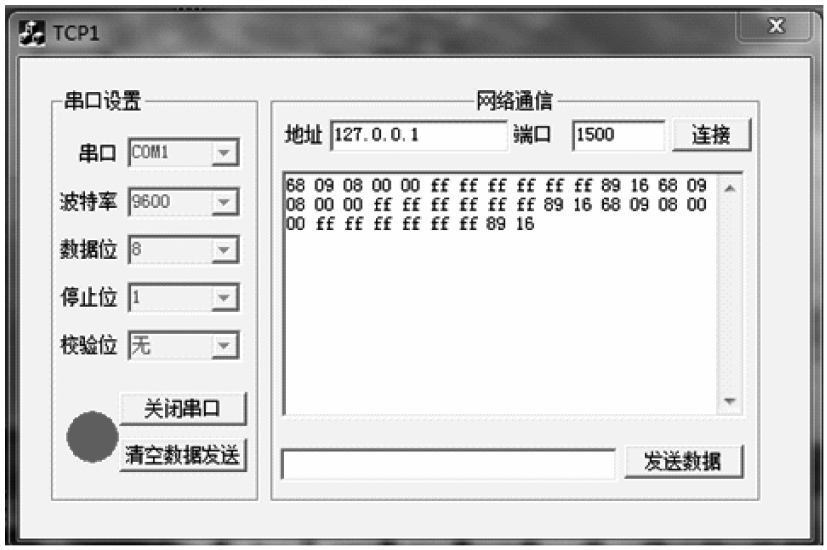


图 5-39　读卡

在服务器端编写程序，根据不同卡号进行相关操作，比如根据卡号向音频设备发出回答响应。具体内容可适当地进行相关开发。

此时，设备可成功连接到服务器，下面可做刷卡-服务器应答实验。

首先将读卡器通过串口连接到服务器串口，单击“注册”，如图 5-40 所示，刷卡，成功后可提示注册成功。

图 5-40　服务器 RFID 注册

注册后服务器数据库可显示如图 5-41 所示信息。

表“RfidUsers”中的数据，位置是“HomeUsers”中、“(Local)\cncc.com”上

UserId	UserName	UserAddress
00 01 02 03 04	霍雷	北科大
00 01 02 03 04	王鲁	信息楼
68 08 05 0a 97	霍磊	北京科技大学
68 08 05 5b 93	韩振	北京科技大学
68 08 05 b2 61	韩振	北京科技大学

图 5-41　服务器数据库信息

然后下次刷卡，服务器根据卡号响应，并发出“你好，＊＊＊”。

5.6　思考题

1）硬件调试、焊接中遇到的问题都有哪些？分别是怎样解决的？

2）对 Mifare S50 射频卡可以进行哪些操作？

3）除了本实验以外，请再思考本实验设备与智能家居系统的一种智能交互方式。

5.7　实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤

（包括软硬件设计方案、软件流程图、硬件原理图等）以及调试过程、心得体会，最后要附上程序，并按要求回答思考题1）、2）、3）。

参 考 文 献

[1] 康东，石喜勤，李勇鹏. 射频识别（RFID）核心技术与典型应用开发案例［M］. 北京：人民邮电出版社，2008.

[2] 周永彬，冯登国. RFID安全协议的设计与分析［J］. 计算机学报，2006，29（4）.

[3] 余雷. 基于RFID电子标签的物联网物流管理系统［J］. 微计算机信息，2006，22（2）.

[4] 闫胜利，袁芳草，冷小冰. Altium Designer 6. x 中文版使用教程——原理图与PCB设计［M］. 北京：电子工业出版社，2007.

[5] 吴晞，马瑞，李星光. RFID系统及其在图书馆中的应用［J］. 图书馆论坛，2005，1.

[6] 张晖，王东辉. RFID技术及其应用的研究［J］. 微计算机信息，2007，23（4）.

[7] 李如年. 基于RFID技术的物联网研究［J］. 中国电子科学研究院学报，2009，4（6）.

[8] 单承赣，单玉峰，姚磊. 射频识别（RFID）原理与应用［M］. 北京：电子工业出版社，2008.

[9] 陈锦斌，林宇洪，邱荣祖. RFID技术在农产品物流系统中应用现状与展望［J］. 物流科技，2013，36（2）.

[10] 无线龙. 高频RFID技术高级教程［M］. 北京：冶金工业出版社，2012.

[11] 李苏剑，张益强. 无线射频识别技术（RFID）理论与应用［M］. 北京：电子工业出版社，2004.

[12] 陈志芳，吴光荣，李昆仑. 一种新型UHF RFID读写模块的研制［J］. 现代电子技术，2013，36（8）.

[13] 田宇，郝强. RFID技术在煤炭企业仓储管理中的应用［J］. 物流技术，2012（5）.

[14] Dobkin D M. The RF in RFID：UHF RFID in Practice［M］. Newnes，2012.

[15] Viani F，Salucci M，Robol F，et al. Design of a UHF RFID/GPS fractal antenna for logistics management［J］. Journal of Electromagnetic Waves and Applications，2012，26（4）.

[16] Lehpamer H. RFID design principles［M］. Artech House，2012.

[17] 袁曜，孟雅娟，高东英，等. RFID技术在血站信息化管理中的应用探索［J］. 中国输血杂志，2012，10.

[18] 赵建光，范晶晶. 基于RFID的智能小区车辆管理系统研究［J］. 河北建筑工程学院学报，2012，30（3）.

[19] 王志良，韩振，朱应剑，等. RFID读写器制作实训教程［M］. 北京：机械工业出版社，2013.

第 6 章　智能家居之服务机器人实验

6.1　实验目的

通过本实验对机器人和安卓的开发有基本了解，同时了解安卓中的 Socket 通信机制和 Wi-Fi 通信等。

6.2　实验内容

实现对机器人的运动控制，包括前进、后退、左转、右转；传感器控制，包括眼睛的亮灭；以及智能控制，包括唱歌、跳舞、控制家电。

6.3　实验所用仪表及设备

1）硬件：装有安卓开发环境的计算机一台、机器人一台、手机一部。

2）软件：JDK、ADT bnudle。

6.4　实验原理

6.4.1　系统硬件环境

1. 整体组成

终端机器人的机械结构如图 6-1 所示。机械结构主要包括底盘、头部和躯干三个部分。底盘主要包括了运动轮系和电动机控制机构、电池放置结构等。躯干部分主要携带 3 个高保真音响，并在机器人前部安装 7 寸触摸显示屏用于交互。头部是终端机器人的重点，其主要包括了头部左右运动、上下运动以及相关传感器。头部左右运动通过头部下方的头部电动机 2 实现、头部上下运动使用头部电动机 1 实现，传感器主要包括头部和颈部的触摸传感以及情感表达的相关 LED。

终端机器人的硬件系统由底层扩展系统、上层 Cortex-A8 硬件平台以及相应的周边模块组成，如图 6-2 所示。机器人具体的功能则是在整个基础上通过安卓系统及应用软件实现。

将硬件系统固定到机械结构上，组装底盘、躯干和头部各部分，实现具体的物联网终端机器人，实际效果如图 6-3 所示。

2. 硬件分系统介绍

（1）系统供电

对于机器人系统而言，由于其可移动性，在供电上采用电池供电。系统中供电最高电压

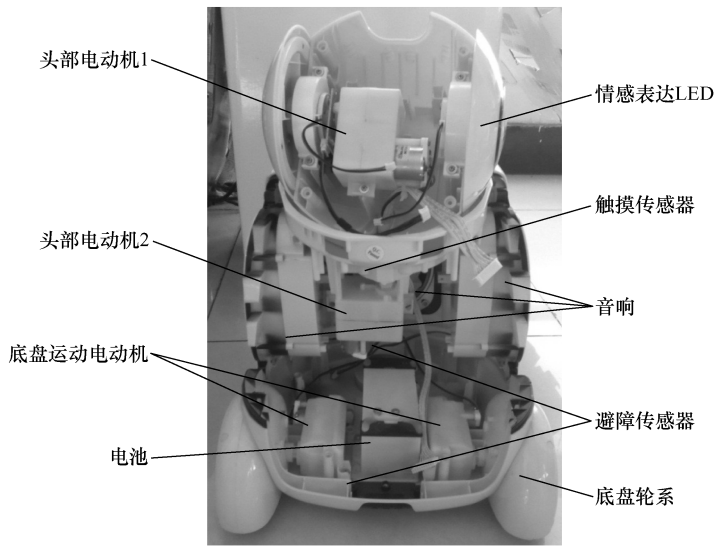


图 6-1 终端机器人机械结构

图 6-2 硬件系统图

为 5V，并存在电动机、触摸显示屏等电流较大的设备，所以选用 7.2V、2300mA、最大电流为 3A 的锂电池。

系统中需要提供的电压为 5V 和 3.3V，其供电结构如图 6-4 所示。电池电源输入系统后一部分用于为电动机运动提供电压，另一部分作为数字电源供给 5VDC-DC。经过转换的 5V 电源一部分给底层扩展系统供电，另一部分则提供给上层 Cortex-A8 平台。各层通过自身的 DC-DC 进行 3.3V 电压的转化。

5V 电源模块为整个系统的数字电源核心，经过的电流较大，因此选用最大电流为 3A 的 LM2596 低压差开关电源芯片。3.3V 电源模块采用 AM1117-3.3 线性稳压电源芯片。电源部分原理图如图 6-5 所示。

（2）主控平台

机器人采用单处理核心，由 Cortex-A8（S5PV210）统筹系统的所有外设，进行设备控制和信息采集。

图 6-3　终端机器人

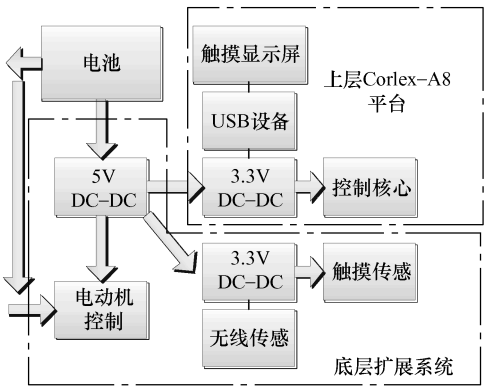


图 6-4　供电结构图

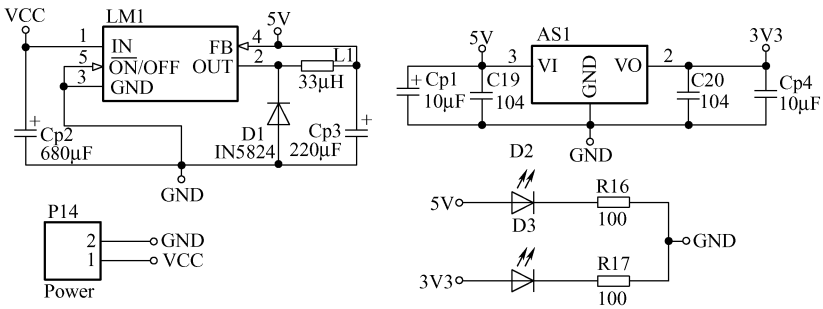


图 6-5　供电系统原理图

整体的系统硬件架构如图 6-6 所示。

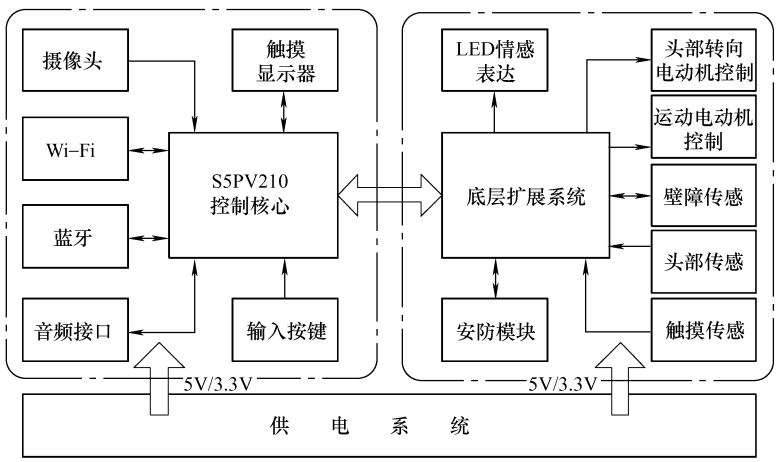


图 6-6　系统硬件架构

底层扩展系统主要用于传感器数据采集、机器人动力系统控制以及情感控制，设备的类型和控制较为简单。而上层 Cortex- A8 平台则是在核心控制器的基础上搭载触摸显示屏、用户按键接口、音频输出以及 USB 类的摄像头、Wi- Fi 适配器、蓝牙适配器。

Mini210s 是一款高性能高性价比的 Cortex-A8 开发板，搭载三星 S5PV210 处理器，运行主频可高达 1GHz，同时集成 PowerVR SGX540 高性能图形引擎。Mini210s 开发板配备了 512MB DDR2 内存和 4GB MLC NAND Flash，其外部应用接口非常丰富，如 WM8960 音频芯片、miniHDMI 高清输出、USB2.0、CMOS 摄像头、SDIO、矩阵键盘等接口，其电源电路也支持低功耗待机省电模式。该款开发板提供开源的安卓 2.3.1 系统源码以及 Linux Kernel 源码，方便用户进行二次开发。同时其还提供方便易使用的开发板烧写工具 Superboot，缩短了系统开发周期。

根据终端系统功能设定要求，Mini210s 开发板的板载资源主要使用电阻触摸显示、USB OTG 接口、音频接口三个部分，其接口布局如图 6-7 所示。电阻触摸显示接口与 7 寸一线精准电阻触摸显示屏连接，实现安卓系统操作界面的显示和触控操作的检测。Wi-Fi、蓝牙、USB 摄像头等设备通过与 USB HOST 接口连接的 USB Hub 完成基于 USB 设备的设备扩展。音频部分采用 I2S0 接口，外接 WM8960 CODEC 解码芯片，提供声音输入和 Speaker 输出（内置 D 类功放，能直接连接 8Ω/1W 扬声器）功能。

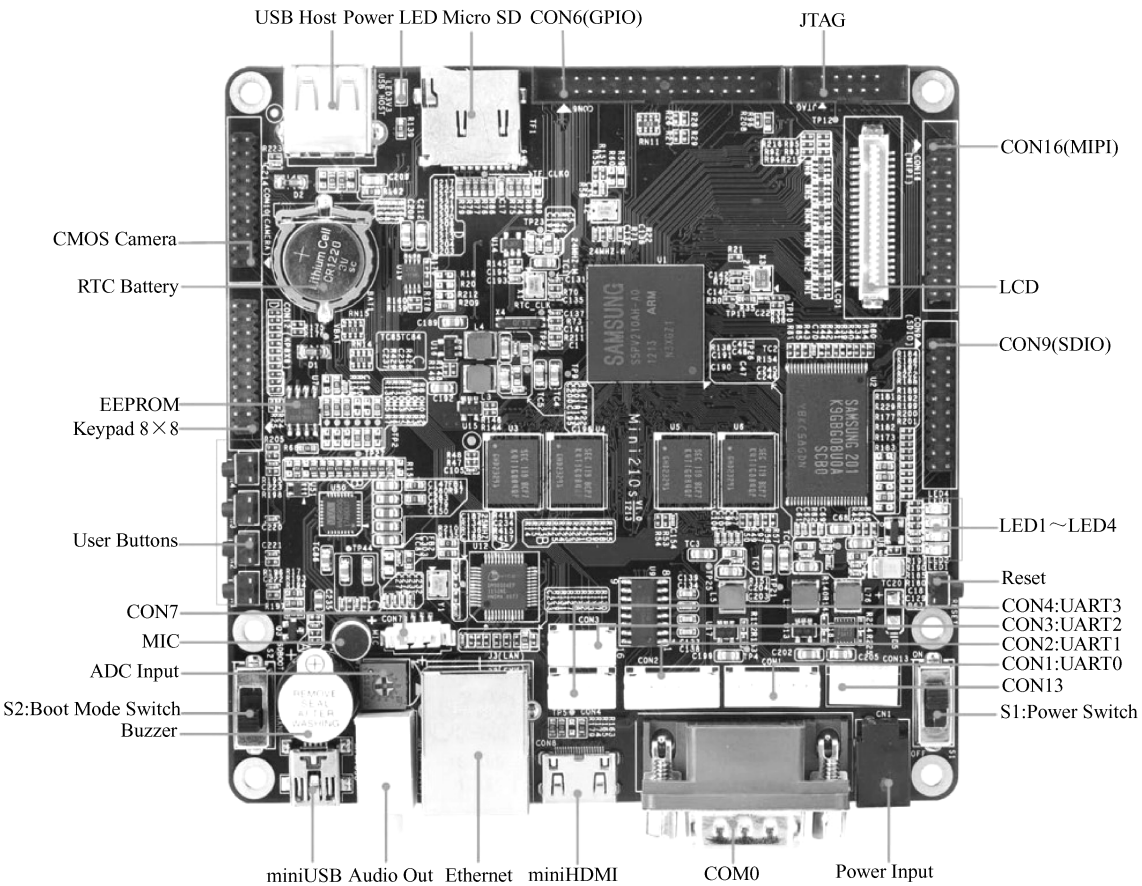


图 6-7　Mini210s 接口布局

（3）接口硬件分系统

接口硬件分系统以 IO 数据总线为核心，综合情感表达、机器人动作控制、数据采

集和无线安防等功能。基于 Arduino 开源平台的思想，硬件底层扩展系统对外提供规范化接口，实现统一化的底层平台，使用任何控制核心与该系统连接即可实现底层设备的控制，可用于可移动物联网节点的快速搭建。以下主要对机器人的动力系统做简单的介绍。

机器人的动力系统通常采用电动机以及舵机等进行实现。舵机是在一个平面上提供一定范围角度（通常在 180°范围内）的旋转，控制方式简单只需要提供固定周期不同占空比的 PWM（Pluse Width Modulation）波即可。普通舵机的力矩较小，力矩较大的舵机价钱偏贵，因此机器人动力系统采用电动机的方式。电动机能够提供正反两个方向的运动，其选型比较灵活，可以根据动力的需求如力矩、速度等进行选配。从控制的角度来讲，单一方向的运转控制比较简单，只需要将电动机连接电源即可。正反同时运转的控制电路则相对复杂一些，主要涉及电源反向电路，最常见的控制方式有继电器和 H 桥方式。

继电器控制思想比较简单，如图 6-8 所示。当继电器 1 和 8 引脚没有电流通过时，电动机的两端 6 引脚连接 +24V，3 引脚连接 GND；当继电器 1 和 8 引脚有电流通过时，产生磁场，吸引导电铁片，使得电动机的接线反转，6 引脚连接到 GND，3 引脚连接到 +24V，从而实现电源的反向。控制电路中 Motor_out1 用于控制电磁铁端是否有电流，即控制方向，Motor1 用于控制电动机的转速。

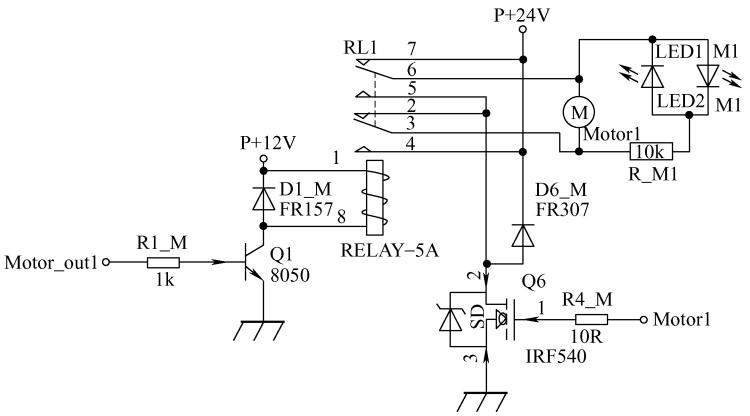


图 6-8　继电器控制电路

H 桥控制原理如图 6-9 所示。H 桥电流由 4 个晶体管或者 CMOS 管（根据具体的工作电压、电流确定）组成，由于其结构类似于字母“H”，所以叫做 H 桥控制电路。当 Q1 和 Q4 导通、Q3 和 Q2 截止时，电流以虚线箭头方向经过电动机；当 Q3 和 Q2 导通、Q1 和 Q4 截止时，电流以实线箭头经过电动机，实现正反两个方向的控制。H 桥控制时需要考虑的重点问题是同臂导通，即 Q1 和 Q2 或者 Q3 和 Q4 同时导通，电源正极和负极相当于直接连接，通过晶体管的电流过大导致晶体管烧坏而整个桥路工作不正常。

机器人使用了 LG9110 电动机控制驱动芯片，该芯片将 H 桥电路内置到单片机 IC 中，使外围器件成本降低，整机可靠性提高。芯片通过两个 TTL/CMOS 兼容电平输入控制电动机输出端的电压方向，具有较大的电流驱动能力，能通过 750 ~ 800mA 的持续电流，峰值电流能达到 1.5 ~ 2.0A。其工作电压在 2.5 ~ 12V 之间，饱和输出压降较小，同时内置的钳位

二极管可释放感性负载的反向冲击电流。LG9110 的器件引脚图和逻辑控制关系如图 6-10 所示。

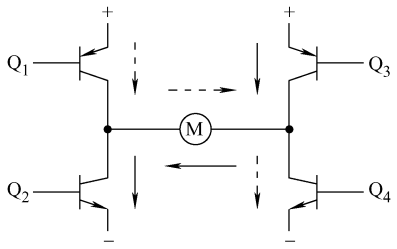
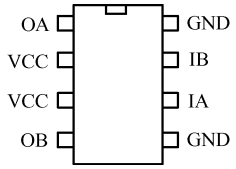


图 6-9 H 桥原理图



IA	IB	OA	OB
H	L	H	L
L	H	L	H
L	L	L	L
H	H	L	L

图 6-10 LG9110 引脚和逻辑控制

与机械结构结合，机器人的动力系统主要由 4 个电动机组成，2 个电动机用于控制机器人的移动，2 个电动机用于机器人的肢体语言的表达。肢体语言主要包括头部的左右转动和抬头低头动作。每个电动机都需要能够正反转动，电路中使用 4 个 LG9110 芯片分别对每个电动机单独控制。正反转的控制需要 8 个 I/O，为了节省 I/O 资源，电路中使用 3 to 8 线译码器 74HC138 将 I/O 资源缩小到 3 个。同时 74HC138 在电路中也起到了电平转化的作用，将 3.3V 电平转化到 5V 电平。在电动机连接端还加入了熔丝以免电流过大导致 LG9110 芯片烧坏。LG9110 应用电路原理图如图 6-11 所示。

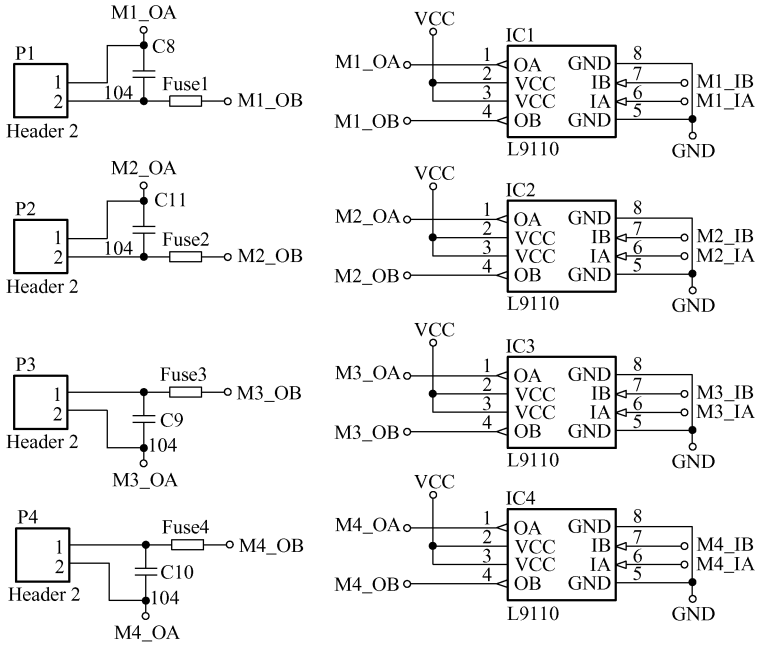


图 6-11 LG9110 应用电路原理图

3. 系统硬件驱动介绍

机器人搭载了安卓 2.3.1 操作系统，其软件系统开发实现主要依托安卓系统完成。安卓系统是移动终端目前最流行的开源操作系统，拥有庞大而又完善的组织架构。安卓平台通过分层结构有效地将底层硬件与上层应用分离，其产品的开发层次清晰，易于实现。正是这种

结构使得安卓系统中的软件开发方式存在 2 种不同的形式：安卓系统开发和安卓应用程序开发。

安卓系统开发属于系统底层开发，主要针对安卓系统本身进行完善和移植。如图 6-12 所示，其开发的工作主要分为 2 个层次：Linux 的驱动和硬件抽象层开发；Libraries 和安卓 Runtime 的开发。第一个层次主要是针对不同的硬件平台实现和移植设备驱动和安卓专有驱动，并通过硬件抽象层向上提供接口。第二层次则主要是在本地编写 C/C + + 代码通过 JNI 向上提供相应的调用接口，为应用层开发提供各种服务。安卓系统开发直接针对安卓系统的源码，在源码的基础上修改和添加程序模块实现对新服务的支持。

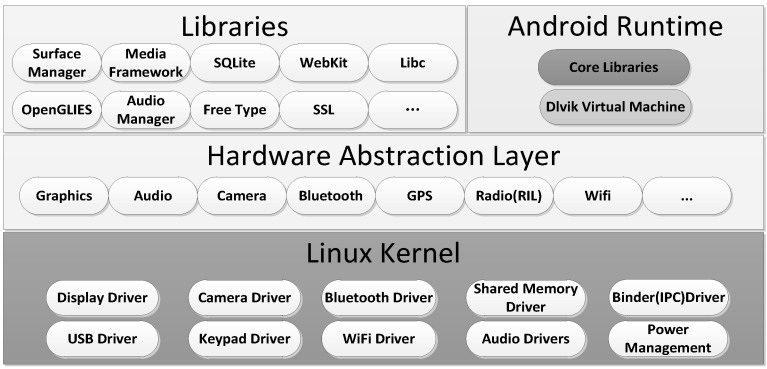


图 6-12　安卓系统开发层次结构

安卓应用开发则是根据系统本身所提供的 API 完成应用 APP 级别的程序开发，使用的 API 属于安卓的 Framework 层，安卓系统的应用开发是安卓得以流行的重要原因，正是其丰富的 API 和用户多种多样的应用需求造就了其应用商店的繁荣。安卓应用开发主要工具是 Google 提供的 Android SDK，采用 Java 语音进行开发，开发环境支持第三方集成环境，官方推荐 Eclipse。通过使用 SDK 自带的移动模拟器，安卓的应用开发可以基本独立于硬件设备完成开发。除了接听和拨打电话以及一些特定硬件功能以为，模拟器能够模拟安卓设备上的典型功能和行为。

驱动开发涉及了安卓系统的 2 种开发方式。整个终端以 Mini210s 开发板为核心，并在其基础上进行电动机驱动、传感器等功能的扩展，因此需要对安卓底层进行开发。友善之臂的 Mini210s 针对安卓系统已经做好了开发板板载资源的相关移植工作，机器人系统底层开发更多地关注于电动机控制、传感器等功能部件的具体实现。在应用层针对智能家居控制功能进行 Java 软件程序的编写，并嵌入云语音识别技术，实现更人性化的交互。

机器人动作、传感器采集、情感表达、无线安防等功能部件都是根据物联网终端机器人的具体需求而进行的扩展，安卓的原生库并不支持此功能，即 Framework 层中不提供相应的 API 和 Service。

如图 6-13 所示，Framework 层中底层硬件 Service 的实现采用自下而上的结构。首先在 Linux 中实现对应功能模块的驱动程序，保证设备的正常运行；然后分别在 HAL、JNI 层建立相应的访问接口；最后在 Framework 层实现本地服务，应用程序即可进行底层硬件的控制。下面详细介绍底层调用的具体实现。

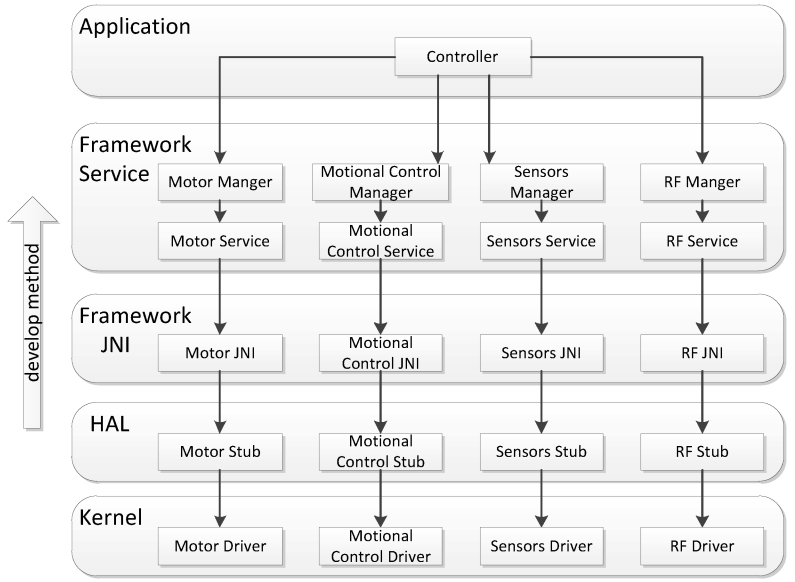


图 6-13 驱动开发结构

6.4.2 软件环境搭建及程序设计

1. 安卓环境搭建

(1) JDK 安装

如果还没有 JDK，可以去

http：//www. oracle. com/technetwork/java/javase/downloads/index. html 下载，然后进行安装。安装完成后设置环境变量步骤如下：

1) 在我的电脑→属性→高级→环境变量→系统变量中添加以下环境变量。

2) JAVA_HOME 值为“ D: \Program Files\Java\jdk1. 6. 0_18 (即安装 JDK 的目录)”。

3) CLASSPATH 值为“. ;% JAVA_HOME%\ lib\ tools. jar;% JAVA_HOME%\ lib\ dt. jar;% JAVA_HOME%\bin;”。

4) Path 在开始追加“% JAVA_HOME%\bin;”。

注意：前面四步设置环境变量对搭建安卓开发环境不是必须的，可以跳过。

安装完成之后，打开 cmd 窗口，输入“java - version”检验是否安装成功，并且可以查看 JDK 的版本信息。出现类似如图 6-14 所示画面表示安装成功。

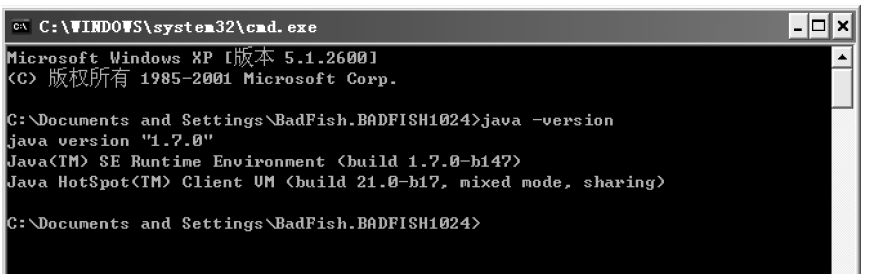


图 6-14 验证 JDK 安装是否成功

（2）安卓集成版开发环境安装

1）如果还没有软件，可以去 http：//developer. android. com/sdk/index. html 下载，将会看到如图 6-15 所示界面。

2）单击“Download the SDK”，将会出现如图 6-16 所示界面，选择需要的版本。

3）单击“Download the SDK ADT Bundle for Windows，”下载、解压，使用时单击 eclipse 即可。

图 6-15　安卓集成版开发环境下载界面

（3）Android SDK 环境变量的配置

新建用户变量，变量名为 Path 变量值为：Android SDK 中的 tools 绝对路径（本书中均以本机为例，D：\Program Files\adt-bundle-windows-x86-20130219\sdk\tools），如图 6-17 所示。

单击“确定”后，重新启动计算机，使新建变量生效。重启计算机以后，进入 cmd 命令窗口，检查 SDK 是不是安装成功。运行“android-h”，如果有类似图 6-18 所示的输出，表明安装成功。

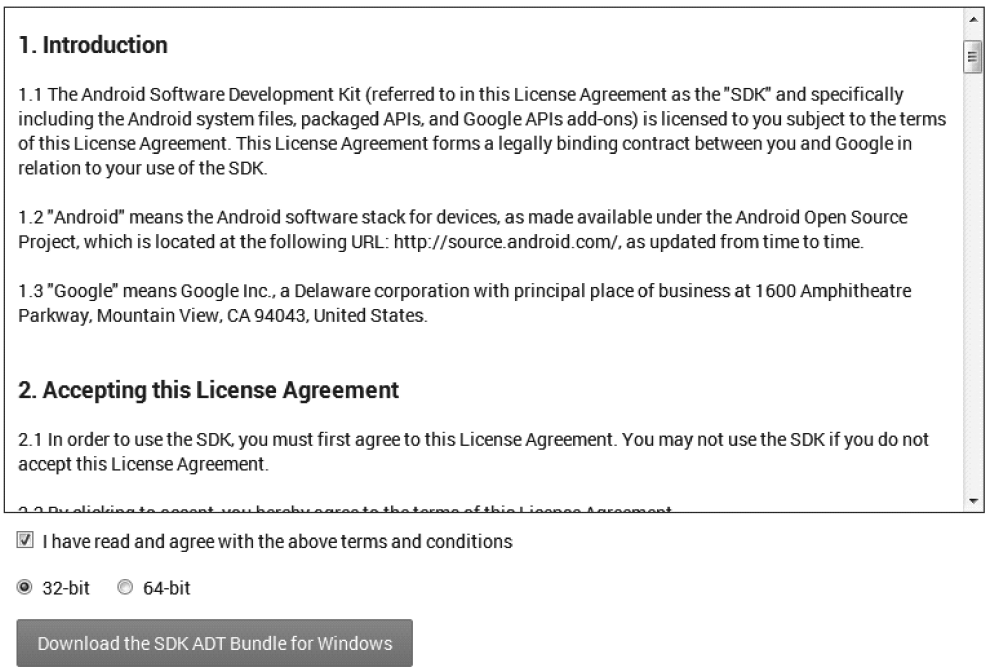


图 6-16　安卓集成版开发环境下载界面

（4）ADB 环境变量的配置

1）添加安卓系统环境变量。在系统变量下单击“新建”按钮，输入环境变量名 Android，将安卓开发工具的路径导入，如图 6-19 所示，然后单击“确定”。

图 6-17　设置安卓 SDK 的环境变量

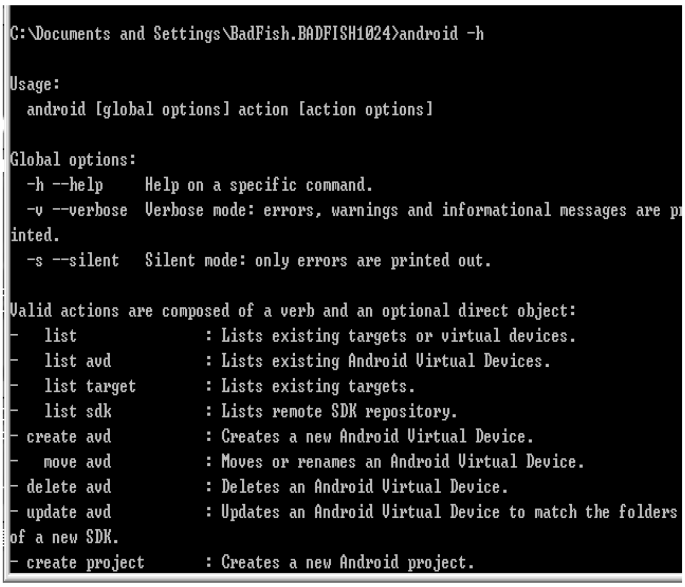


图 6-18　验证安卓 SDK 是否安装成功

3 个变量值：

1. .；
2. D:\Program Files\adt-bundle-windows-x86-20130219\sdk\tools；
3. D:\Program Files\adt-bundle-windows-x86-20130219\sdk\platform-tools；

图 6-19　新建安卓系统环境变量

注意：一定要用英文输入法。

2）再新建系统变量 Path，在其中的变量值里面填写“% Android%”，如图 6-20 所示，单击“确定”完成。

图 6-20　新建 Path 系统环境变量

打开 cmd 窗口，输入“adb”，检验是否配置成功，出现类似如图 6-21 所示画面表示安装成功。

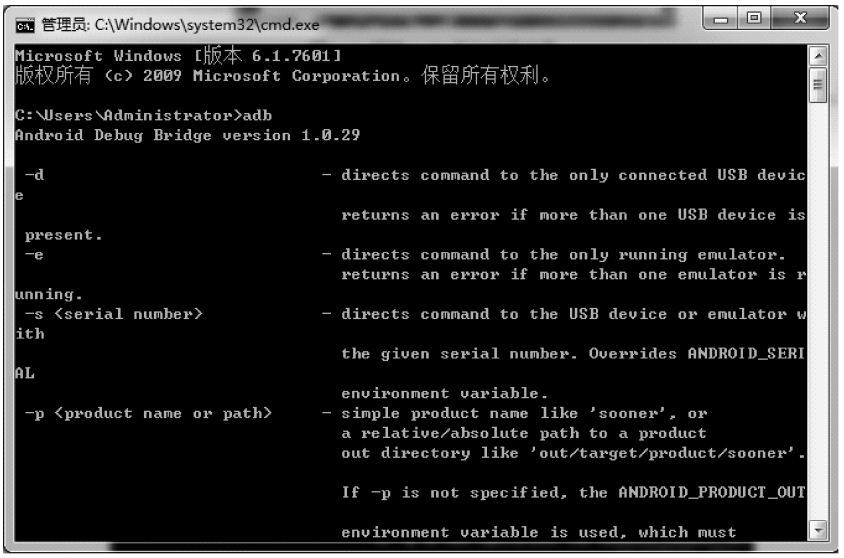


图 6-21　adb 环境配置成功

（5）创建 AVD

为使安卓应用程序可以在模拟器上运行，必须创建 AVD。

1）在 Eclipse 中。选择 Windows > Android Virtual Device Manager。

2）单击左上角的“Android Virtual Devices”，在右侧单击“New”，如图 6-22 所示。

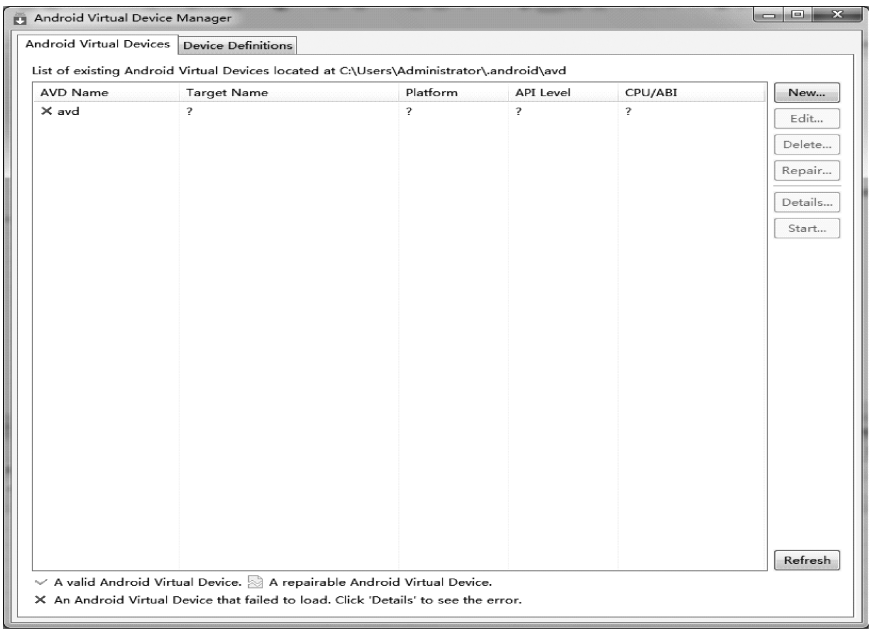


图 6-22　创建 AVD 示意图

3）填入 Name，选择 Target 项的 API，SD Card 项大小任意，Skin 项随便选，这里给出示意图如图 6-23 所示。

4）设置好后，单击“OK”，即可完成创建 AVD，如图 6-24 所示。

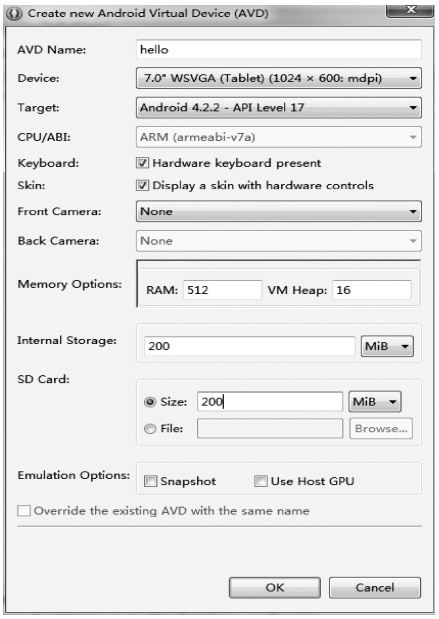


图 6-23　创建 AVD 示意图

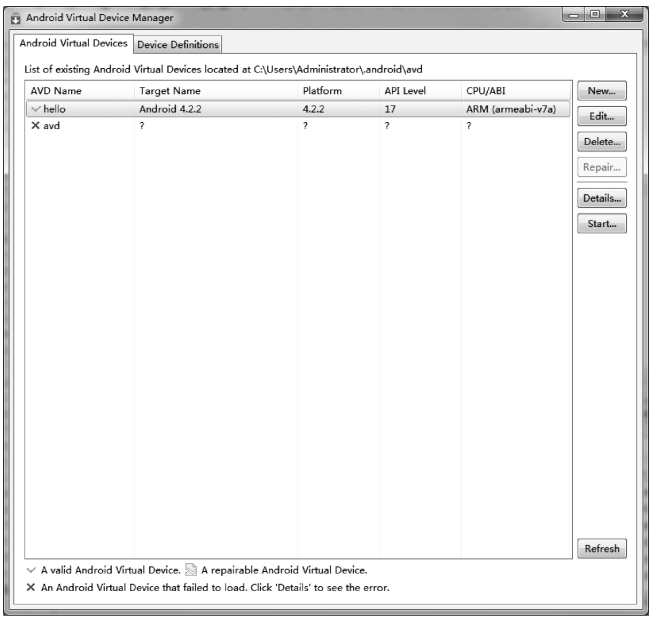


图 6-24　创建 AVD 成功示意图

（6）创建工程项目 HelloWorld

通过选择 File → New → Project 菜单，建立新项目“Android Project”，如图 6-25 所示。

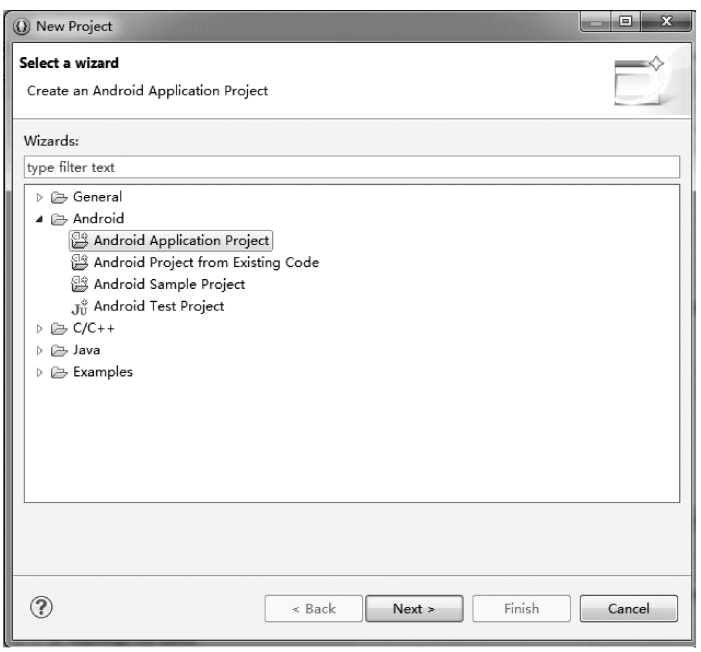


图 6-25　新建 Android 项目示意图一

然后填写必要的参数，如图 6-26 和图 6-27 所示。

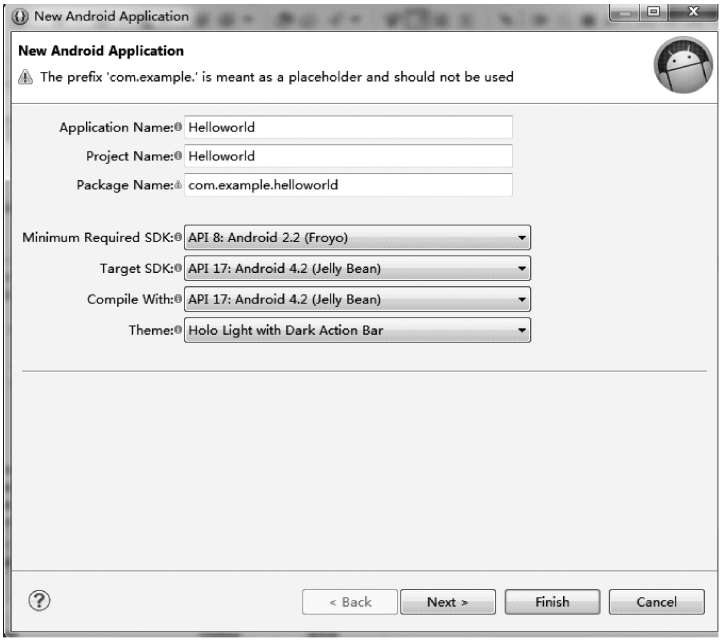


图 6-26　新建安卓项目示意图二

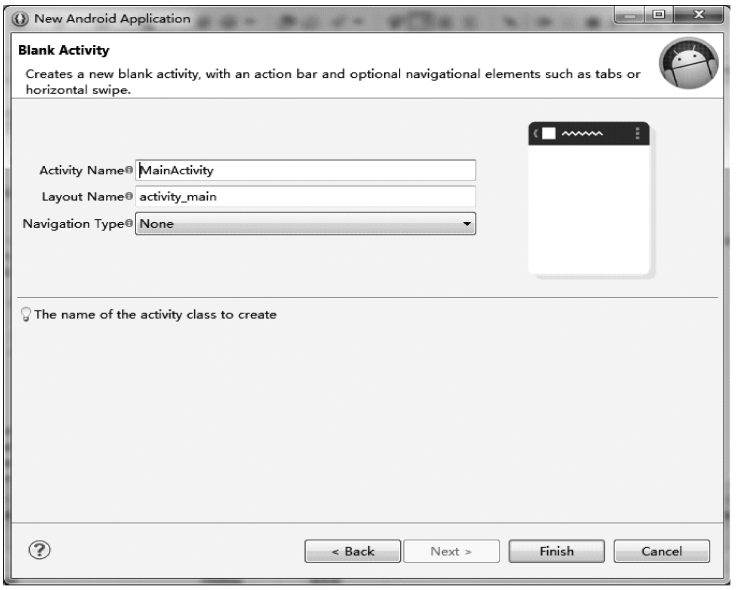


图 6-27　新建安卓项目示意图三

相关参数的说明如下：

1）Project Name：包含这个项目的文件夹的名称。

2）Package Name：包名，遵循 Java 规范，用包名来区分不同的类是很重要的，此处用的是 com. mxw。

3）Activity Name：这是项目的主类名，这个类将会是安卓的 Activity 类的子类。Activity 类是一个简单的启动程序和控制程序的类，它可以根据需要创建界面，但不是必需的。

4）Application Name：一个易读的标题在你的应用程序上。

单击“Finish”后，单击 Eclipse 的 Run 菜单选择 Run as---Android Application.

运行成功后会有 Android 的模拟器界面，如图 6-28 所示。

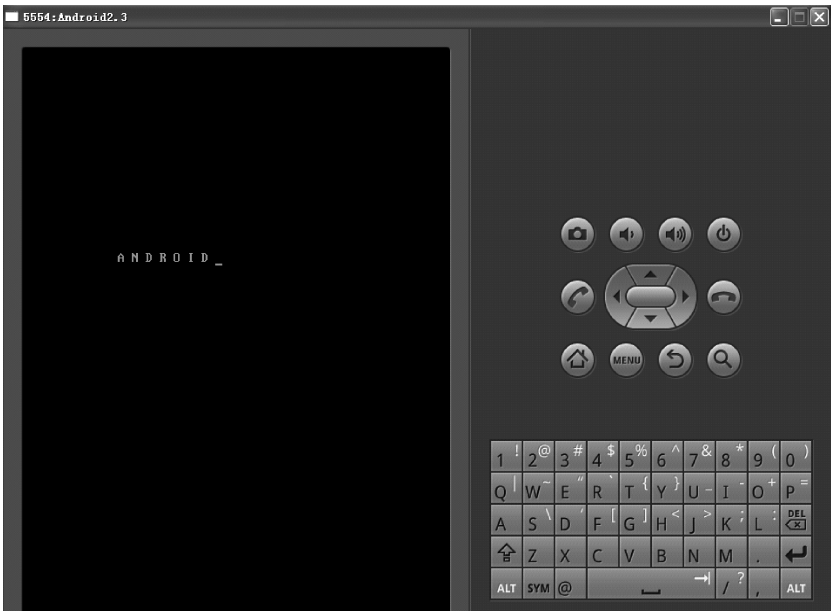


图 6-28　程序运行过程示意图

2. Socket 简介

（1）什么是 Socket

它实际上是网络通信的一种接口，基于不同的协议，有各种不同的 Socket，比如有基于 TCP 的 Socket 和基于 UDP 的 Socket 以及基于蓝牙协议的 Socket，Android 中使用的是 Java 的 Socket 模型。

Socket 在计算机行业通常称为“套接字”，用于描述 IP 地址和端口，是一个通信链的句柄。这是一个比较抽象的概念。计算机是拥有端口的，每一个端口都可以有一个应用程序来进行通信的使用，比如：80 端口是 HTTP 所使用的端口，21 端口是 FTP 所使用的端口，端口是计算机和外界通信的接口，这些接口都是逻辑的接口，端口号取值的范围是零 ~256 乘以 256 减 1，1024 以下的端口都是操作系统使用的保留端口，以上的端口可以自由地来使用，不要与其他应用程序的端口冲突。

应用程序可以通过“套接字”向网络发送请求或者应答网络的请求，这时就把 Socket 分成了两部分，一部分是服务器端的 Socket，这个 Socket 主要用来接收来自网络的请求，它一直监听在某一个端口上。一部分是客户端的 Socket，这个 Socket 主要用来向网络发送数据。

（2）Socket 通信模型（见图 6-29）

（3）UDP 和 TCP

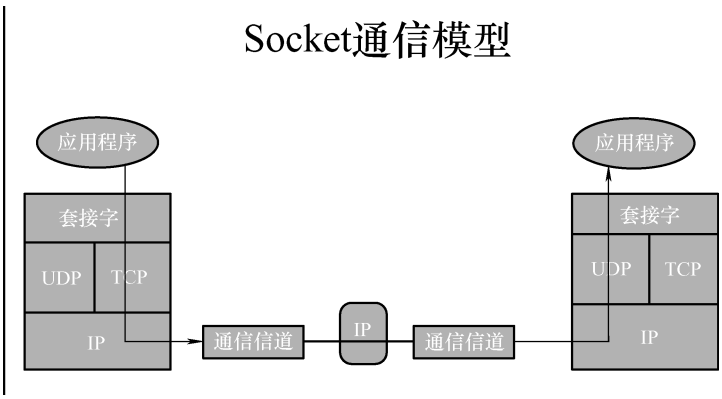


图 6-29 Socket 通信模型

UDP 和 TCP 是互联网使用最广的两种协议，都是基于 IP 的协议。第一个区别是 UDP 是一个不太靠谱的协议，UDP 把数据都打成数据包，数据包上自带通信地址，也就是说要把这个数据包发送到网络上的哪一个地址，通过网络把这个数据包发送出去，至于这个数据包是否发送到目的地，是否服务器端接收到了这个数据包，这个协议并不保证。TCP 发送数据时要求接收方接收到数据之后给一个回应，也就是是否收到了，TCP 可靠一些，当我们发送一些比较重要的数据的时候一般都使用 TCP。另外一个区别是 UDP 发送的一个数据包它的容量是有限的，而 TCP 则没有这样一个限制。并不是说 UDP 一定就不如 TCP，在不同的领域有不同是使用，UDP 好处是速度相对快些，而 TCP 相对慢些。

（4）Socket 通信流程

应用程序通过“套接字”也就是 Socket 可以选择这两种协议当中的一种，你可以选择用 UDP 发送数据，也可以选择用 TCP 发送数据，数据发送出去通过“通信信道”也就是 IP 的基础网络，来到服务器端（接收端），就可以接收到数据了。发送数据时用 UDP，接收时也要用 UDP，发送数据时用 TCP，接收时也要用 TCP，在发送时指定接收端的 IP 地址和端口号就可以了，究竟数据包或数据是如何发送的，框架已经封装好了，暂且不用去关心它。TCP 通信示意图如图 6-30 所示。

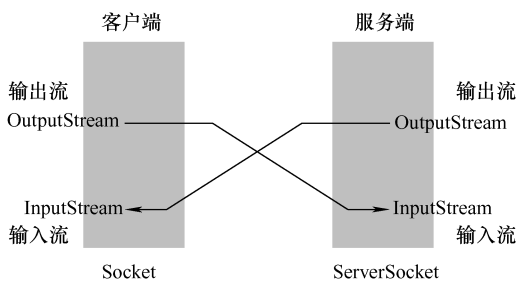


图 6-30 TCP 通信示意图

（5）编程步骤

1）创建服务器的步骤：

① 指定端口实例化一个 ServerSocket。

② 调用 ServerSocket 的 accept 方法以在等待连接期间造成阻塞。

③ 获取位于该底层 Socket 的流以进行读写操作。

④ 将数据封装成流。

⑤ 对 Socket 进行读写。

⑥ 关闭打开的流。

2）创建客户端的步骤：

① 通过 IP 地址和端口实例化 Socket，请求连接服务器。

② 获取 Socket 上的流以进行读写。

③ 把流包装进 BufferedReader/PrintWriter 的实例。

④ 对 Socket 进行读写。

⑤ 关闭打开的流。

当要监控多个客户端时，可使用类 ExecutorService。其对象获取方法：Executors. newCachedThreadPool()；调用该对象的 execute（Runnablecommand）即可，在未来某个时间执行给定的命令。该命令可能在新的线程、已入池的线程或者正调用的线程中执行，这由 Executor 决定。

3. Wi-Fi 连接介绍

大致可以分为四个主要的类 ScanResult、WifiConfiguration、WifiInfo WifiManager。

（1）ScanResult

主要是通过 Wi-Fi 硬件的扫描来获取一些周边的 Wi-Fi 热点的信息。在进行 Wi-Fi 搜索时，一般会搜到这些信息，首先是接入点名字、接入点信息的强弱，还有接入点使用的安全模式（如 WPA、WEP）。

打开这个类，我们可以看到以下几个信息。

1）BSSID，接入点的地址，这里主要是指小范围几个无线设备相连接所获取的地址，比如说两台笔记本通过无线网卡进行连接，双方的无线网卡分配的地址。

2）SSID，网络的名字，当搜索一个网络时，就是靠这个来区分每个不同的网络接入点。

3）Capabilities，网络接入的性能，这里主要是来判断网络的加密方式等。

4）Frequency，频率，每一个频道交互的频率数。

5）Level，等级，主要用来判断网络连接的优先数。

这里只提供了一个方法，就是将获得信息编程字符串 toString()。

（2）WifiConfiguration

在连通一个 Wi-Fi 接入点时，需要获取到一些信息。大家可以与有线的设备进行对比一下。

这里的数据相对来说比较复杂。

（1）6 个子类

WifiConfiguration. AuthAlgorthm 用来判断加密方法。

WifiConfiguration. GroupCipher 获取使用 GroupCipher 的方法来进行加密。

WifiConfiguration. KeyMgmt 获取使用 KeyMgmt 进行。

WifiConfiguration. PairwiseCipher 获取使用 WPA 方式的加密。

WifiConfiguration. Protocol 获取使用哪一种协议进行加密。

WifiConfiguration. Status 获取当前网络的状态。

（2）设置 WifiConfiguration 的属性

WifiConfiguration. AuthAlgorthm 设置加密方法。

可选参数：LEAP 只用于 leap，OPEN 被 WPA/WPA2 需要，SHARED 需要一个静态的

WEP Key。

WifiConfiguration. GroupCipher 使用 GroupCipher 的方法来进行加密。

可选参数：CCMP，TKIP，WEP104，WEP40。

WifiConfiguration. KeyMgmt 键管理机制（keymanagerment），使用 KeyMgmt 进行。

可选参数：IEEE8021X、NONE、WPA_EAP、WPA_PSK。

WifiConfiguration. PairwiseCipher 设置加密方式。

可选参数：CCMP、NONE、TKIP。

WifiConfiguration. Protocol 设置一种协议进行加密。

可选参数：RSN、WPA。

WifiConfiguration. Status 设置当前网络的状态。

可选参数：CURRENT，DISABLED，ENABLED。

（3）WifiInfo

在 Wi-Fi 已经连通以后，可以通过这个类获得一些已经连通的 Wi-Fi，获取当前链接的信息，这里信息就比较简单了，简单介绍一下获得周围 Wi-Fi 信息的方法：

1）getBSSID( ) 获取 BSSID；

2）getDetailedStateOf( ) 获取客户端的连通性；

3）getHiddenSSID( ) 获知 SSID 是否被隐藏；

4）getIpAddress( ) 获取 IP 地址；

5）getLinkSpeed( ) 获得连接的速度；

6）getMacAddress( ) 获得 Mac 地址；

7）getRssi( ) 获得 IEEE 802. 11n 网络的信号；

8）getSSID( ) 获得 SSID；

9）getSupplicanState( ) 返回具体客户端状态的信息。

（4）WifiManager

WifiManager 用来管理 Wi-Fi 连接，这里已经定义好了一些类，可以供使用。

这里来说相对复杂，里面的内容比较多，但是通过字面意思，还是可以获得很多相关的信息。这个类里面预先定义了许多常量，可以直接使用，不用再次创建。

下面简单介绍一下管理 Wi-Fi 连接的方法：

1）addNetwork（WifiConfiguration config）通过获取到的网络的连接状态信息，来添加网络；

2）calculateSignalLevel（int rssi，int numLevels）计算信号的等级；

3）compareSignalLevel（int rssiA，int rssiB）对比连接 A 和连接 B；

4）createWifiLock（int lockType，String tag）创建一个 Wi-Fi 锁，锁定当前的 Wi-Fi 连接；

5）disableNetwork（int netId）让一个网络连接失效；

6）disconnect( ) 断开连接；

7）enableNetwork（int netId，Boolean disableOthers）进行一个连接；

8）getConfiguredNetworks( ) 获取网络连接的状态；

9）getConnectionInfo( ) 获取当前连接的信息；

10）getDhcpInfo( ) 获取 DHCP 的信息；
11）getScanResulats( ) 获取扫描测试的结果；
12）getWifiState( ) 获取一个 Wi-Fi 接入点是否有效；
13）isWifiEnabled( ) 判断一个 Wi-Fi 连接是否有效；
14）pingSupplicant( )，判断当前网络是否能连通；
15）ressociate( ) 即便连接没有准备好，也要连通；
16）reconnect( ) 如果连接准备好了，进行连通；
17）removeNetwork( ) 移除某一个网络；
18）saveConfiguration( ) 保留一个配置信息；
19）setWifiEnabled( ) 让一个连接有效；
20）startScan( ) 开始扫描；
21）updateNetwork （WifiConfiguration config）更新一个网络连接的信息。

6.5　实验步骤

6.5.1　机器人运动控制

1）搭建好安卓开发环境，打开 Eclipse 开发环境，等待软件启动，启动过程如图 6-31 所示。

2）软件启动完成后，出现如下界面，即为 Eclipse 的编程环境，如图6-32 所示。

3）下面将存在的安卓工程文件夹导入到 Eclipse 开发环境中，机器人控制端的源程序文件夹为 Controlclient 文件夹，先复制项目目录到 Workspace 目录中，再执行以下操作，如图 6-33 所示。

4）选择“Existing Projects into Workspace”，如图 6-34 所示。

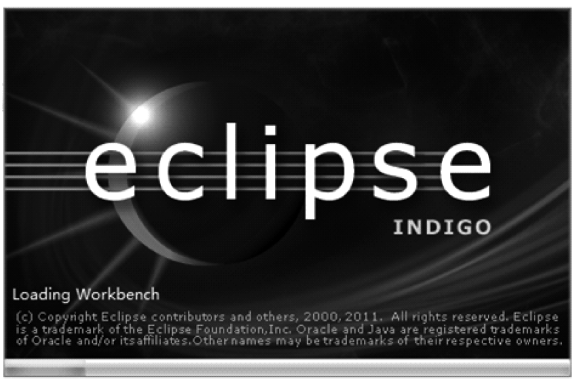


图 6-31　软件启动过程示意图

5）浏览并选定项目所在目录，项目文件夹为“Workspace”下的“Controlclient1 静态”，如图 6-35 所示：

注意：如果之前没将项目复制到 Workspace 目录，那么这时也可以通过此窗口中的这个选项让 Eclipse 帮你把项目复制到 Workspace 中去，如图 6-36 所示。接着单击“Finish”就完成导入了。

6）打开自己的安卓端移动设备（手机或者平板电脑），将手机的 Wi-Fi 打开，连接至当前可用的一个 Wi-Fi 网络中。然后在设置中选择应用程序，选择开发，然后选中 USB 调试，再将你的连接至计算机的 USB 端口。

7）用 USB 线连接手机到计算机，一般情况会自动安装驱动，如果无法安装驱动可以去下载一个安装助手，让它帮手机自动安装驱动。

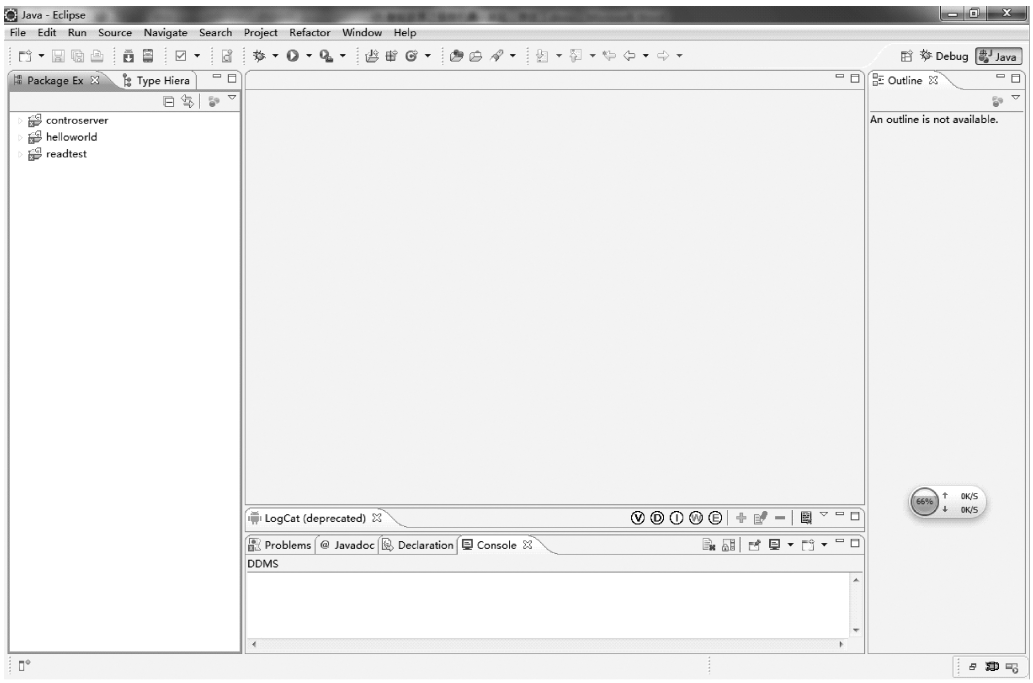


图 6-32 Eclipse 编程环境

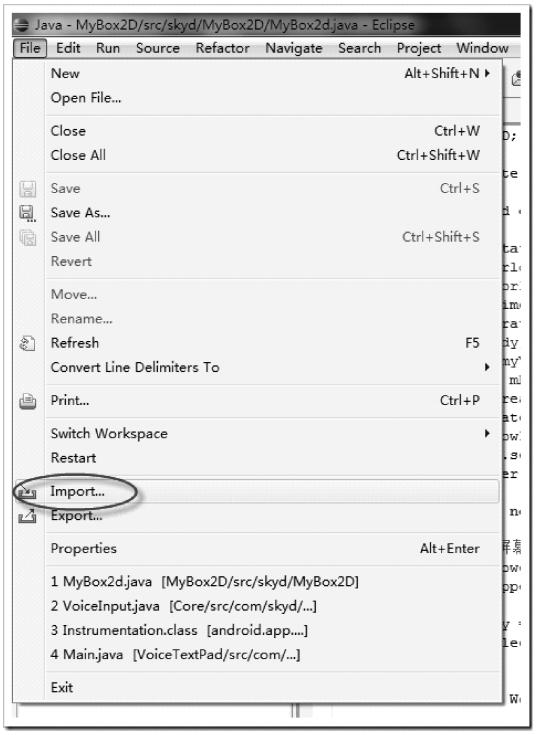


图 6-33 Import 工程示意图

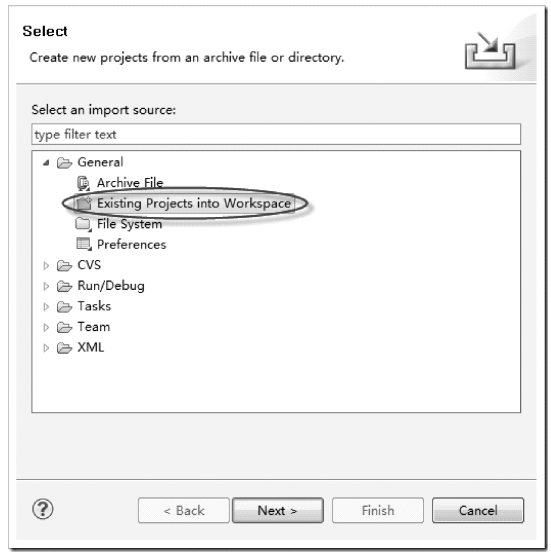


图 6-34 选择已存在工程示意图

图 6-35　打开程序示意图

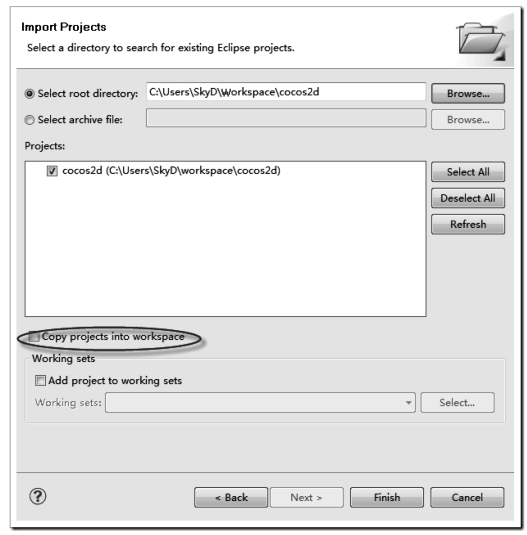


图 6-36　复制工程到 Workspace 示意图

8）在 Eclipse 中单击 Controlclient 项目中会有所有的工程列表，右键单击 Controlclient 工程项目，在级联菜单中找到 run as，并选中 run as 的级联菜单下的 Android Application 选项，即可很快地在真机上看到应用程序的运行结果。如果确实无法进入真机的调试状态，可以进入 Eclipse 配置选项。在 Eclipse 中，单击鼠标右键选中项目 Run As→Run Configuractions→Target→Manual→Run，弹出 Android Device Chooser 对话框，选中真机作为 Target，单击“OK”，即可将项目运行到真机上如图 6-37 和图 6-38 所示。

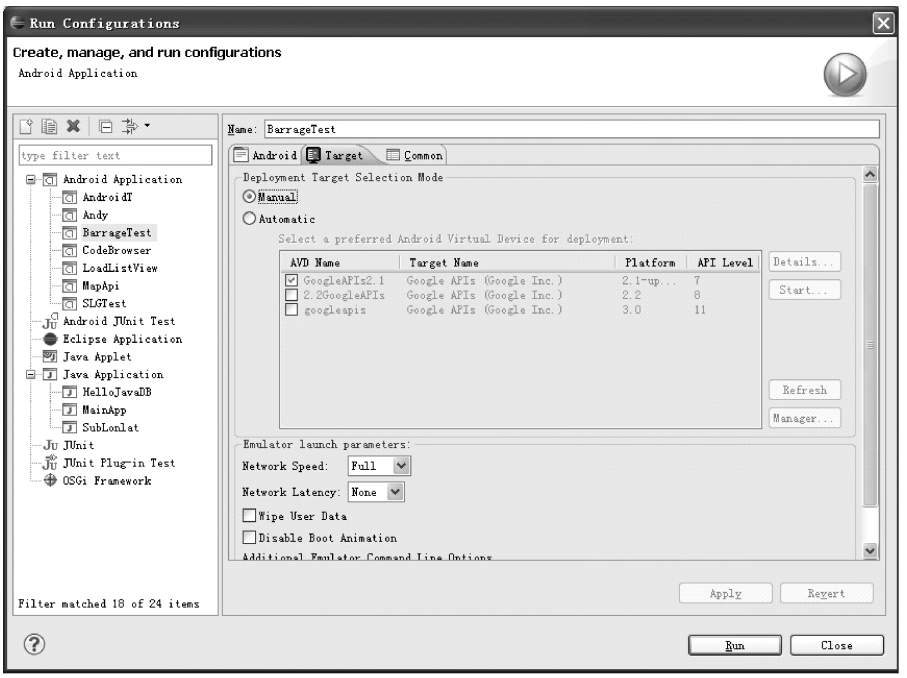


图 6-37　选择真机运行示意图

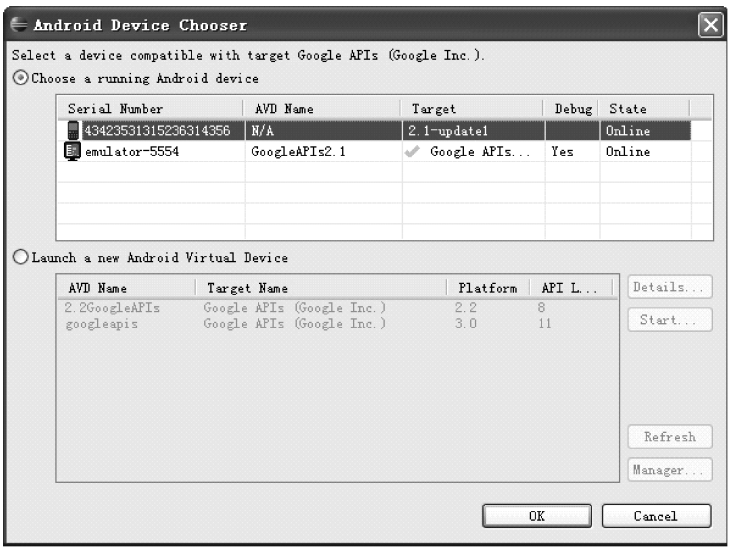


图 6-38 选中真机示意图

9）然后会在你的安卓设备中看到如下界面，如图 6-39 所示。

10）打开机器人背后的电源开关，等待机器人的系统启动完毕，出现如下界面，如图 6-40 所示。

图 6-39 Controlclient1 程序界面示意图

图 6-40 机器人前方平板界面示意图

11）打开桌面的快捷方式 Controlserver 应用程序，弹出以下界面，如图 6-41 所示。

若在桌面上未能找到 Controlserver 应用程序，请单击主界面右侧的 ，打开“所有应用程序”的界面，在下面的界面中找到 Controlserver 程序，如图 6-42 所示。

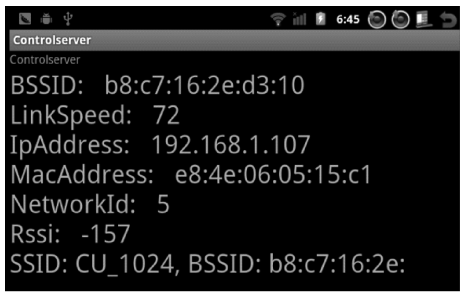


图 6-41　Controlserver 程序界面示意图

12）在你的安卓设备中的 Controlclient 程序界面中找到“目标 IP 地址”一项，将机器人中的 Controlserver 程序中 IpAdress 项的 IP 地址，如图 6-43 所示，输入到其中，如图 6-44 所示。

请确保此程序显示的 IP 地址与目标 IP 在同一个网段下（如 192. 168. 1. xxx），若未在一个网段下，请将此安卓设备的 Wi-Fi 连接到正确的无线网络中，并重复之前的步骤重新打开此应用程序。

图 6-42　机器人前方平板所有应用程序界面示意图

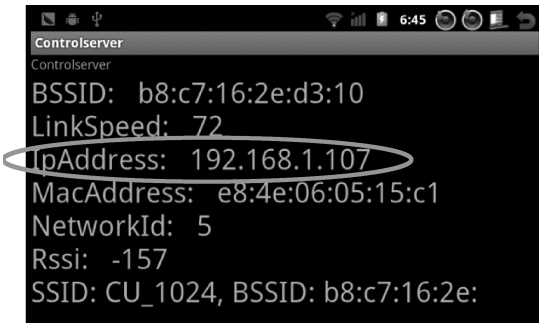


图 6-43　Controlserver 界面示意图

图 6-44　Controlclient IP 填写示意图

13）若要对机器人的运动进行控制，请单击“控制小优整体运动”下的左转、右转、前进、后退、停止等相应的按键，如图 6-45 所示。

图 6-45 Controlclient 控制按钮示意图

14）若要对机器人的情感表达进行控制，请单击“控制小优面部运动”下的眼睛、耳朵、嘴巴等相应的按键。同时，单击“唱歌”、“跳舞”等按键可使得机器人做相应的综合动作，如图 6-46 所示。

图 6-46 Controlclient IP 设置完成整体界面示意图

6.5.2 机器人智能家居控制

1）在上一节实验的基础上，打开机器人系统中的“所有应用程序”界面，找到“智能家居”应用程序并打开，如图 6-47 所示。

注意：请确保机器人所连接的无线网络处在智能家居实验室内的无线路由范围中。

2）打开程序后弹出以下界面，单击“控制界面”按钮，进入下一个程序界面，如图 6-48 所示。

3）在图 6-48 所示控制界面中直接单击“顶灯”按钮，将打开智能家居实验室内的顶灯，再次单击将关闭，如图6-49所示。

4）单击如图 6-48 所示控制界面中的“电视”按钮，弹出如图 6-50 所示界面，可对电视进行各项操作。

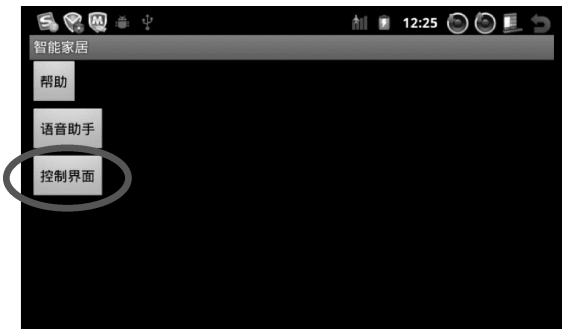


图 6-47　智能家居首界面示意图

图 6-48　控制界面示意图

图 6-49　控制界面示意图

5）返回到主界面，单击“空调”按钮，弹出如图 6-51 所示界面，可对空调进行各项操作。

图 6-50 控制电视界面示意图

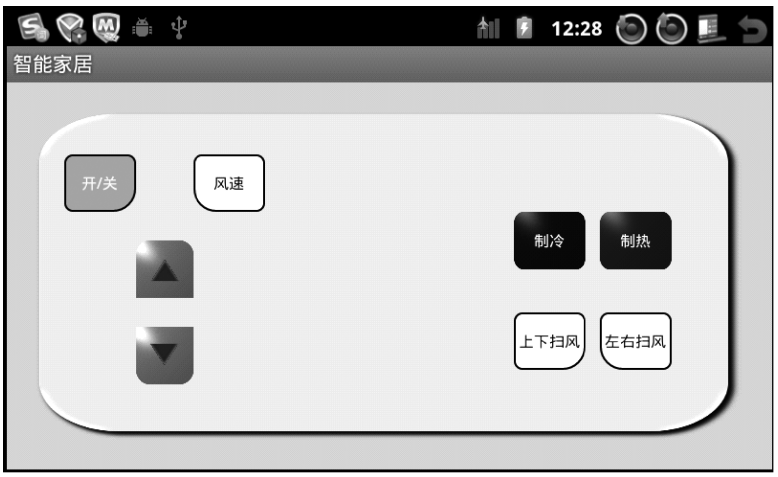


图 6-51 控制空调界面示意图

6.6 思考题

1）如果想要添加机器人的一个功能，譬如停止唱歌，那么 Controlclient 和 Controlserver 程序应该分别做怎样的修改？

2）如果想要同时控制两台机器人，则程序应该怎样修改？

3）机器人的通信协议不按照 Constant 中规定的方式，请想一个更为安全、规范的协议。

6.7 实验报告内容及要求

实验报告内容应包括实验目的、实验内容、实验设备、实验步骤以及心得体会，并按要

求完成程序的设计和修改。

参 考 文 献

[1] 何立民. 从嵌入式系统视角看物联网 [J]. 单片机与嵌入式系统应用, 2010 (10).

[2] 王茜. Android 嵌入式系统架构及内核浅析 [J]. 开发与应用, 2011 (4).

[3] 董士海. 人机交互的进展及面临的挑战 [J]. 计算机辅助设计与图形学学报. 2004 (01).

[4] 高云峰, 吴秀芬. 服务机器人视觉系统模块化研究综述 [J]. 机械设计与制造, 2010 (2).

[5] 杨丰盛. Android 技术内幕系统卷 [M]. 北京: 机械工业出版社, 2011.

[6] 金泰延, 宋亨周, 朴知勋. Android 框架揭秘 [M]. 北京: 人民邮电出版社, 2012.

[7] 叶炳发. Android 操作系统移植及关键技术研究 (硕士学位论文) [D]. 广州: 暨南大学, 2010.

[8] 马越. Android 的架构与应用 (硕士学位论文) [D]. 北京: 中国地质大学 (北京), 2008.

[9] 姚星吴, 刘卫国. Android 的架构与应用开发研究 [J]. 计算机系统应用, 2011 (11).

[10] 陈璟, 陈平华, 李文亮. Android 内核分析 [J]. 现代计算机, 2009 (11).

[11] Daniel P. Bovet, Marco Cesati. 深入理解 Linux 内核 (影印版) [M]. 南京: 东南大学出版社, 2006.

[12] 岳传真. Android 系统移植和应用程序开发 (硕士学位论文) [D]. 上海: 复旦大学, 2010.

[13] 周剑秋. ARM-Linux 的底层 IO 操作的实现研究 [J]. 机电产品开发与创新, 2009, 22 (4).

[14] 王志良, 漫谈物联网 [J]. 科技与企业, 2011 (5): 12-13.

[15] 王田苗, 陈殿生, 殷兰兰. 服务机器人辅助老年人生活的新模式与必要性 [J]. 机器人技术与应用, 2010 (1).

[16] Ostrander J. Android UI Fundamentals: Develop and Design [M]. Peachpit Press, 2012.

[17] 林汝泽, 徐媛媛, 方凯, 等. 基于 HTTP 协议的 Android 手机数据同步实现 [J]. 信息通信, 2013 (1).

[18] 罗鹏. 基于 ZigBee 技术的手机智能家居系统设计 [J]. 中国新通信, 2013 (9).

[19] 高闯, 李海平, 王国栋, 等. 基于 android 平台的智能家居手机客户端软件设计 [J]. 电子测试, 2013 (12).

[20] 解仑, 王志良, 李华俊. 双足步行机器人制作技术 [M]. 北京: 机械工业出版社, 2008.

第 7 章　虚拟管家实验

7.1　实验目的

1）了解智能家居网关服务系统的结构组成、工作流程和设计原理。
2）掌握网关服务的调用方法，实现智能家居管理系统客户端的功能设计。
3）掌握虚拟管家系统，实现语音提示功能。

7.2　实验内容

1）创建新客户端实例。
2）编写 VC 的上位机界面与服务器端进行 Socket 通信。
3）实现语音识别功能。
4）完成智能家居管理的语音提示功能。

7.3　实验所用仪表及设备

1）硬件：Web 服务器、数据库服务器、数据采集服务器。
2）软件：Win7、Microsoft SQL Server 2008、Microsoft Visual Studio 2010、Active X_Installer。

7.4　实验原理

7.4.1　智能家居网关服务系统

在物联网家庭电器、物联网家庭无线中继器、无线传感器等多种硬件系统的配合下，智能家居网关服务系统实现了智能家居中的三大功能：家居设备信息的传输和控制；基于 Web 服务、HTTP 和 TCP 方式的内外网访问控制服务的发布；家庭信息的智能控制。

智能家居网关服务系统既具备网关的所有智能家居功能，分模块和高速运行智能家居管理软件，同时具备可扩展性，对于新设备、新功能可以通过无线或有线方式接入该系统，从而实现物联网智能家居管理系统丰富的功能。

7.4.2　网关服务系统构成

1. 网关服务系统整体结构

网关服务工作框图如图 7-1 所示。

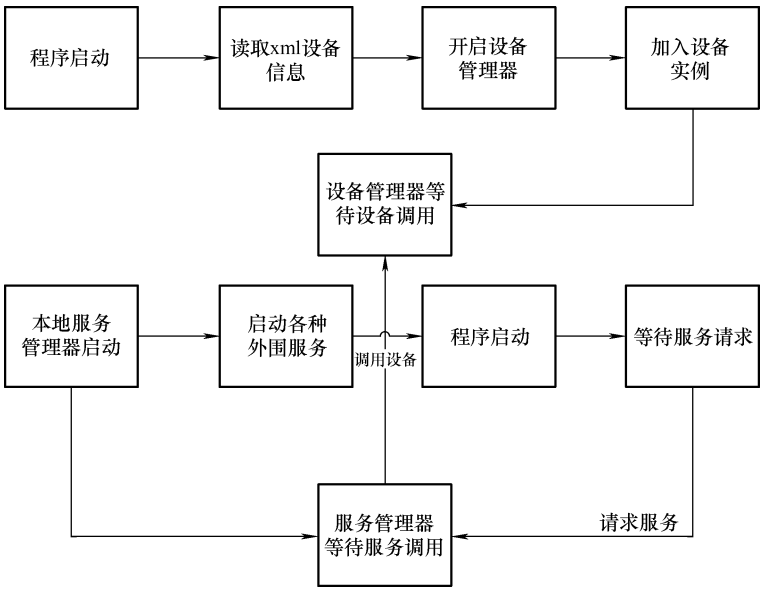


图 7-1　网关服务总体工作框图

网关服务系统的流程如图 7-2 所示。

2. 网关服务系统功能说明

程序启动后，首先要读取设备信息。设备信息包括了设备基本信息、设备操作通道信息、设备操作信息和设备状态信息。初始化与设备信息读取的流程如图 7-3 所示。

设备管理与调用的流程如图 7-4 所示。

7.4.3　网关服务发布

管理软件的模块有很多种，包括 BasicGatewayService、RestfulService、AuthedBasicGatewayService. cs、RemoteSpeechService、SpeechService、VoiceService 等。它们分别发布不同访问方式的服务。

1. BasicGatewayService

BasicGatewayService 是一个 WebService 服务，它是利用微软的 WCF 通信框架发布的。WebService 是工业标准，它能够对自己进行自描述。也就是说，服务端不需要向客户端做任何说明，客户端可以根据这个服务的发布地址，自己写出服务调用代码。

2. RestfulService

RestfulService 是一个 HTTP 风格的调用服务，它是为了应对一些客户端无法生成调用代码或者是有的客户端根本就不能进行开发而设计的。它的显著特点是基于 HTTP 调用服务，因此只要在浏览器输入一段地址就可以访问调用服务。它的 HTTP 调用 API 如下：

RestFul 的 HTTP 调用就是用 HTTP 方式请求一个网络地址，服务器会返回一个操作结果。

HTTP 方式的基地址：http：//115. 25. 48. 254：8902/RestfulService/。

HTTP 方式访问的地址是：基地址 + 相对地址。其中相对地址的 {} 括起来的部分使用真实的数据替换掉。

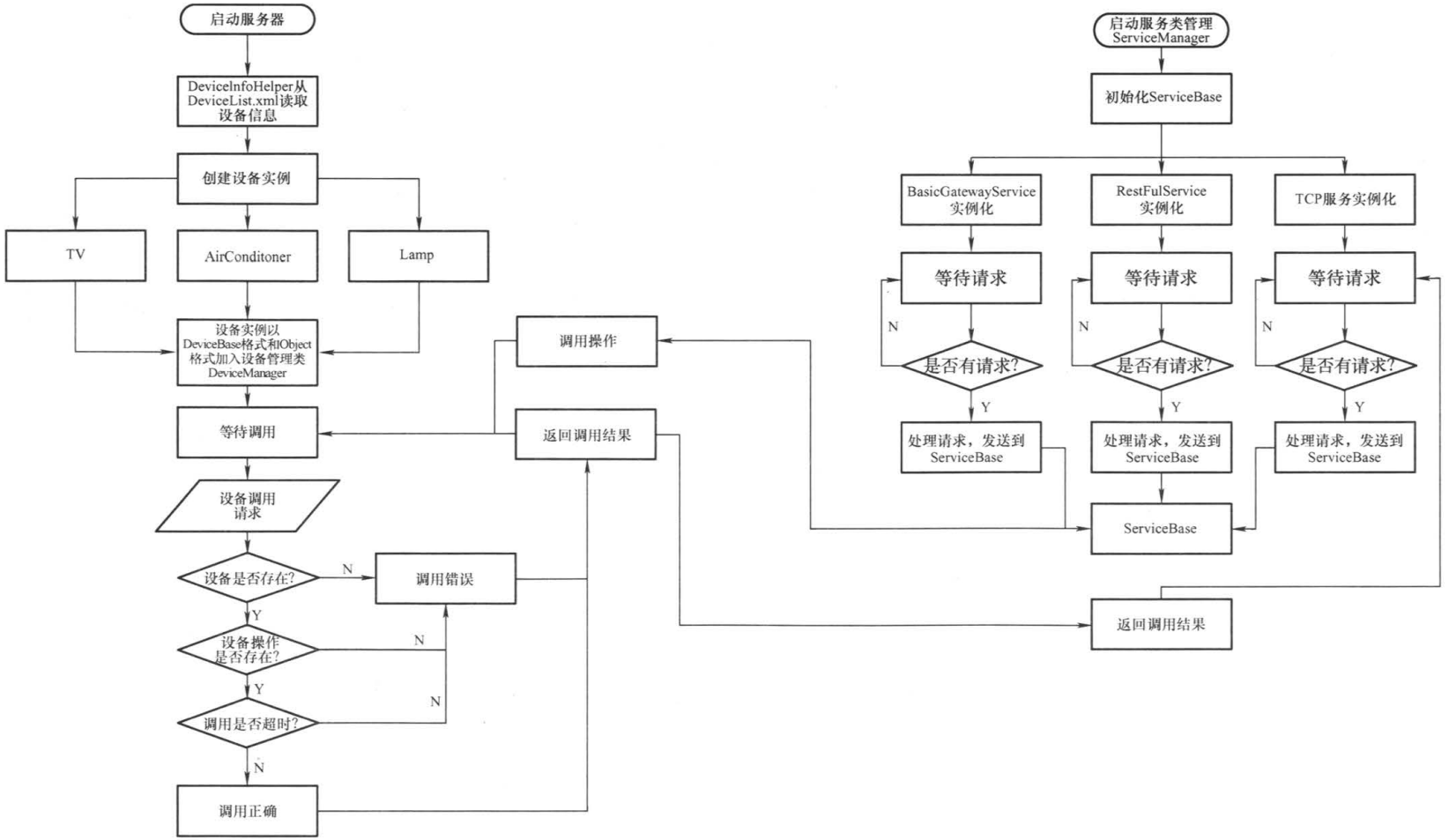


图 7-2　网关服务系统流程图

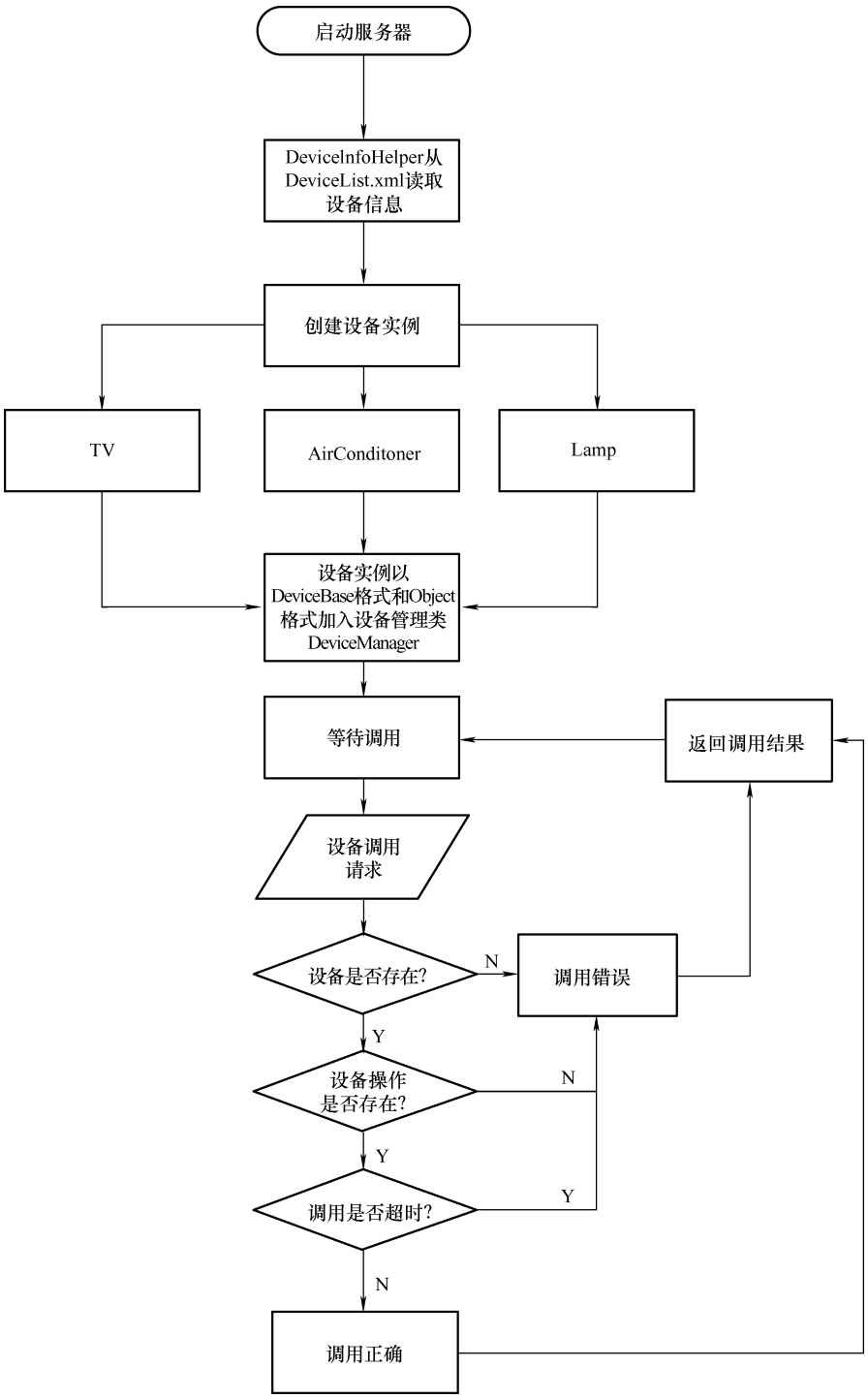


图 7-3　初始化与设备信息读取流程图

相对地址及说明见表 7-1。

其中，每个接口返回的数据格式都是 json 格式，方便了利用 JS 脚本进行解析的过程。

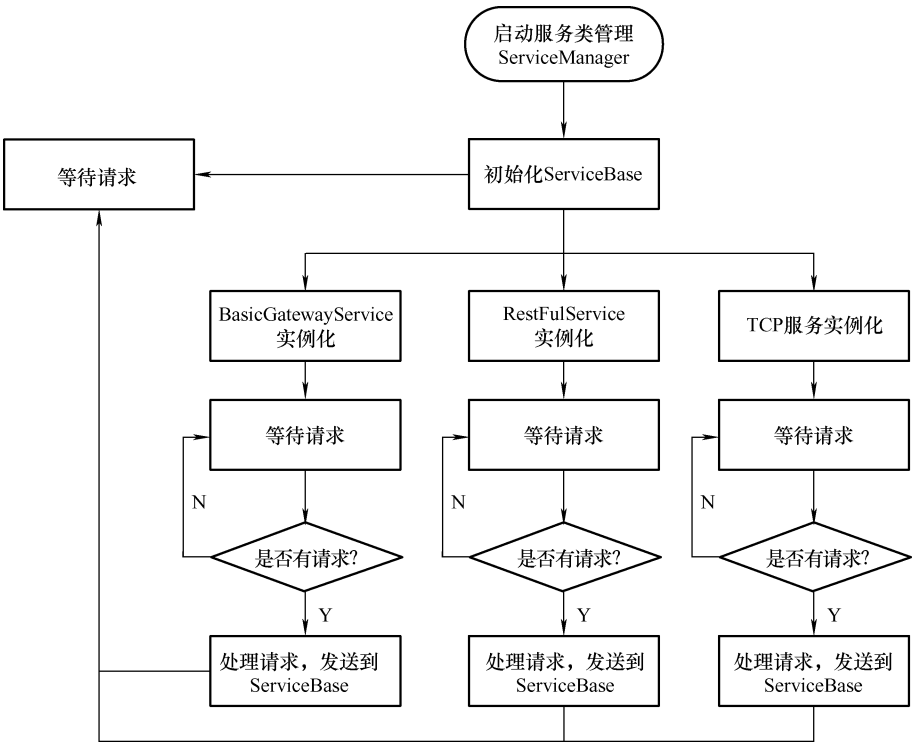


图 7-4　设备管理与调用流程图

表 7-1　相对地址及说明表

相对地址	参　数	返回值	说　明
Gateway? clientId = {clientId}	clientId：客户端 ID	InvokeResult GatewayInfo	得到网关信息
Devices/{deviceId}? clientId = {clientId}	deviceId：设备 ID；clientId：客户端 ID	InvokeResult DeviceInfo	获得指定设备的信息
Devices/List? clientId = {clientId}	clientId：客户端 ID	InvokeResultDictDeviceInfo	获得设备列表
Devices/Count? clientId = {clientId}	clientId：客户端 ID	InvokeResultInt	获得设备数量
Devices/{deviceId}/OperationList? clientId = {clientId}	deviceId：设备 ID；clientId：客户端 ID	InvokeResultDictAgentOperStatus	获得设备的操作列表
Devices/{deviceId}/{operationName}? clientId = {clientId}	deviceId：设备 ID；clientId：客户端 ID operationName：操作名	InvokeResult AgentOperStatus	获得设备的某一个操作信息
Devices/{deviceId}/Call? operationName = {operationName}&operParams = {operParams}&clientId = {clientId}	deviceId：设备 ID；clientId：客户端 ID operationName：操作名 operParams：设备操作参数，目前只接受一个参数	InvokeResultObject	调用设备操作，得到操作结果
Devices/{deviceId}/Picture/? clientId = {clientId}	deviceId：设备 ID；clientId：客户端 ID	字节流形式的图像字符串	得到设备图像
Devices/{deviceId}/Photo? clientId = {clientId}	deviceId：设备 ID；clientId：客户端 ID	字节流形式的图像字符串	摄像头拍照

3. AuthedBasicGatewayService

AuthedBasicGatewayService 是一个基于 C#的用户认证调用接口，它实现了服务调用的安全和权限。每个客户端请求服务前必须进行登录动作，没有登录不能操作。同时服务端根据用户名确定该客户是否有调用设备操作的权限。它仅仅是多了一个登录接口，其他的与 BasicGatewayService 一致。

4. VoiceService

VoiceService 是接收处理语音识别结果的服务。它是一个基于 TCP 的服务。家居控制中的交互方式有很多种，其中一种是利用语音控制。语音识别使用的是在线识别，识别结果是说话内容的文本。这个服务能够接收文本格式的识别内容。由于用户说话内容不确定，识别内容是随机的，这个服务会请求模糊识别的代码进行判断。

5. SpeechService 和 RemoteSpeechService

SpeechService 和 RemoteSpeechService 分别是本地语音合成和远程语音合成的服务。本软件自带语音合成，能够读出设备的调用信息。当语音合成功能需要和软件本身分开，处于两个不同的程序中时（例如，想使用另一台计算机读出管理软件的活动时），这个程序能够把要合成的语音文本传送出去。

7.4.4　服务发布方法

本系统的网络服务访问接口（BasicGatewayService、RestfulService、AuthedBasicGatewayService 等）都是通过微软的 WCF 框架发布的。这些服务都是通过 ServiceManager 发布的。发布代码如下：

BasicGatewayService 发布

```
//服务合约的细节
//这本来是不需要设置的，但是服务传输了图像，造成一次性传输的信息量过大，不设置缓存长度，会出现缓存不够长，而导致通信错误。
WSHttpBinding NoneSecurity = new WSHttpBinding(); //没有安全措施的服务
        NoneSecurity.Name = "NoneSecurity"; //binding 的名字
        NoneSecurity.MaxBufferPoolSize = 2097152; //最大缓存池大小
        NoneSecurity.MaxReceivedMessageSize = 2097152; //最大消息长度
        NoneSecurity.UseDefaultWebProxy = false; //不使用代理
        NoneSecurity.ReaderQuotas.MaxArrayLength = 2097152; //最大序列长度
        NoneSecurity.ReaderQuotas.MaxStringContentLength = 2097152; //最大字符串长度
        NoneSecurity.Security.Mode = SecurityMode.None;
//安全行为服务地址。这个端口号和 IP 地址一定要和实际计算机的一致。
//端口号和地址定义在 GatewayInfo.xml 文件里
        Uri baseAddress = new Uri("http://" + ServiceBase.Instance.GatewayInfo.ServerAddr + ":" + ServiceBase.Instance.GatewayInfo.ServerPort + "/Gateway/");
//把服务托管到 host 里，这时的服务只是个框架，是个容器，里面的内容是空的。
        ServiceHost host = new ServiceHost(new BasicGatewayService(),baseAddress);
```

```
            //添加服务行为
    //设置服务的描述文件（wsdl 文件），可以通过 HTTP 方式获得
            ServiceMetadataBehavior metaBehaviour = new ServiceMetadataBehavior();
            metaBehaviour.HttpGetEnabled = true;
            host.Description.Behaviors.Add(metaBehaviour);
            ServiceMetadataBehavior BasicGatewayMetaBehavior = new
ServiceMetadataBehavior();
            BasicGatewayMetaBehavior.HttpsGetEnabled = true;

            //生成服务终结点。这是真正的服务。
             ServiceEndpoint BaseGatewayPortal = new ServiceEndpoint(ContractDescription.
GetContract(typeof(OpenService.IBasicGatewayService)),NoneSecurity,new EndpointAddress
("http://115.25.48.98:8902/BasicGateway/"));
            // ServiceEndpoint GatewayPortal = new
ServiceEndpoint(ContractDescription.GetContract(typeof(Services.WCF.IGatewayService)),
NoneSecurity1,new
EndpointAddress("http://115.25.48.98:8902/Gateway/"));
            //添加服务终结点。也就是把服务真正地添加到了 host 里
            host.AddServiceEndpoint(BaseGatewayPortal);
            //host.AddServiceEndpoint(GatewayPortal);
            //开启服务，以便客户端访问
            host.Open();
    //通过浏览器访问 http://115.25.48.98:8902/BasicGateway/?wsdl，如果看到返
回的 xml 文档，说明服务开启成功了。
```

7.4.5 客户端调用网关服务

对于 WCF 的客户端来说，WCF 服务就像是一个 Web Service 一样，在 Visual Studio 2008 以上的开发环境中，所有 WCF 服务的连接都是由客户端的服务代理（WCF Service Proxy）来运行，开发人员不用花费太多心思在通信上，而 WCF Service Proxy 在 Visual Studio中被称为服务引用（Service Reference）。

在 Visual Studio 中加入 WCF 的服务引用时，Visual Studio 会自动帮助开发人员完成一些必要工作（例如组态创建以及产生 Service Proxy 等），开发人员只需要在代码中取用 WCF Service Proxy 对象即可。

7.4.6 虚拟管家系统

本系统采用连续播放图像的方式制作动画，模拟虚拟管家的说话动作。

在用户进入系统时，将进行自我介绍，并发出语音。语音内容可以自行设置，如“您好，我是您的虚拟管家小丽，很高兴为您服务。”

在利用网关客户端连接上服务器之后，虚拟管家将进行提示，“你好，客人，欢迎来到

智能空间实验室。”

此后，用户可以利用 GatewayClient 与服务器交互。

用户进入系统并连接网关，对以上任一设施进行操作时，都将进行语音提示。

7.4.7　语音合成服务

语音合成，又称文本到语音转换（Text to Speech）技术，能将任意文字信息实时转化为标准流畅的语音朗读出来，它涉及声学、语言学、数字信号处理、计算机科学等多个学科技术，是中文信息处理领域的一项前沿技术。目前，许多学习软件都是用了 TTS（Text-To-Speech）技术。利用 TTS 可以实现真人发音、控制音量、速率等功能。

Windows XP 系统要想实现语音识别，需要安装 Microsoft Speech API 5.1 开发包。而 Microsoft Speech API 5.1 只支持英文文本朗读，如果需要朗读中文，需要安装微软的 TTS 语音包。Vista、Win7 下自带语音合成引擎和语音库，不需要安装上述文件。微软公司开发的 TTS 文字朗读引擎可以把计算机中出现的文字转换成自然流畅的语音输出，TTS 语音包下载地址如下：

http://download.microsoft.com/download/speechSDK/SDK/5.1/WXP/EN-US/Speechsdk51LangPack.exe

7.5　实验步骤

7.5.1　建立空白解决方案

本系统会有多个项目，为了让各项目之间的关系反映在目录结构上所以要建立一个解决方案，步骤如下：

1）建立解决方案：选择文件→新建项目→其他项目类型→Visual Studio 解决方案→空白解决方案。

2）在解决方案里添加第一个项目：建立空白解决方案后，在 VS2010 右边的解决方案资源管理器中，可以看到刚建立的解决方案，单击选中它，然后单击鼠标右键选择“添加”→新建项目→Visual C#→ASP.NET Web 应用程序，把名称命为：Web，单击确定后就在解决方案里建立了一个 Web 项目。

3）在解决方案里添加第二个项目：在解决方案里加入第一个项目后，再加入第二个项目，第二个项目的添加与第一个项目的添加稍有不同。添加第二个项目的步骤是：文件→新建项目→Visual C# →类库。选择“类库”以后，把名称命名为：BLL，在最下面的“解决方案”里选择“添入解决方案”，意思就是把新建的这个 BLL 项目，添入到建立的 KaiBo 解决方案里。刚才建立的两个项目就在解决方案里很好地组织起来了。

4）在解决方案里添加其他的项目：添加两个项目后，为解决方案添加更多的项目就只需选中解决方案，单击鼠标右键选择“添加”→新建项目→Visual C#→类库。再自己命名项目名称就可以了。

7.5.2　系统操作过程

1）安装开发环境 Visual Studio 2010，并将开发语言设置为 C#。

安装成功后，打开开发环境，如图 7-5 所示。

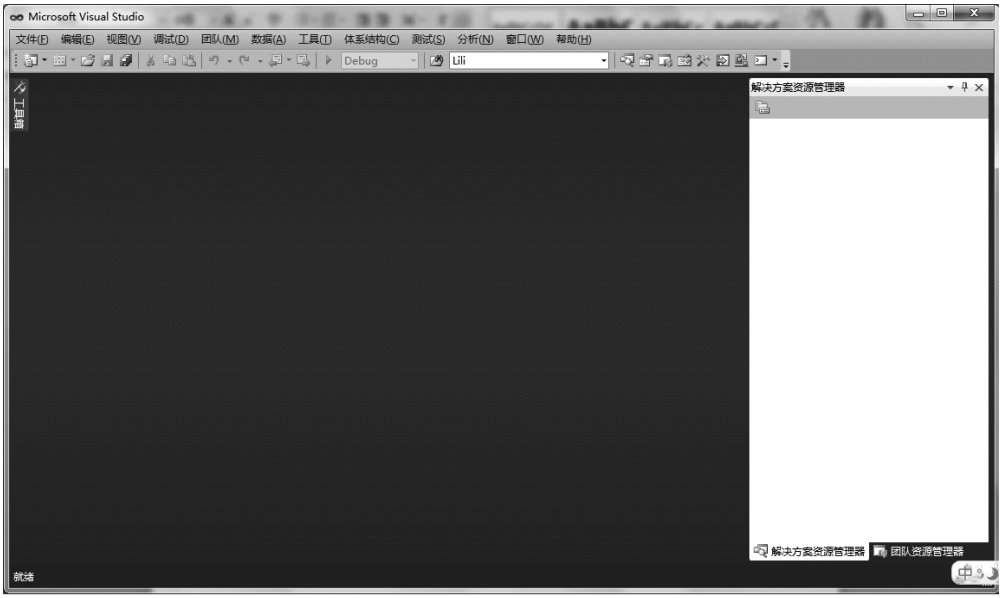


图 7-5　Visual Studio 2010 界面

2）测试语音合成功能。

在 Win7 系统下，打开控制面板→轻松访问→语音识别→文本到语音转换，可以看到语音合成的设置界面，如图 7-6 所示。

图 7-6　语音合成设置

选择能够朗读中文的语音角色，例如“ Microsoft Lili”，单击“预听语音”，听取朗读结果。记下这个角色名备用。

3）打开 Visual Studio2010，打开 GatewayClient 解决方案。

单击文件→打开，找到 GatewayClient 所在目录，选择 GatewayClient 解决方案，如图 7-7 所示。

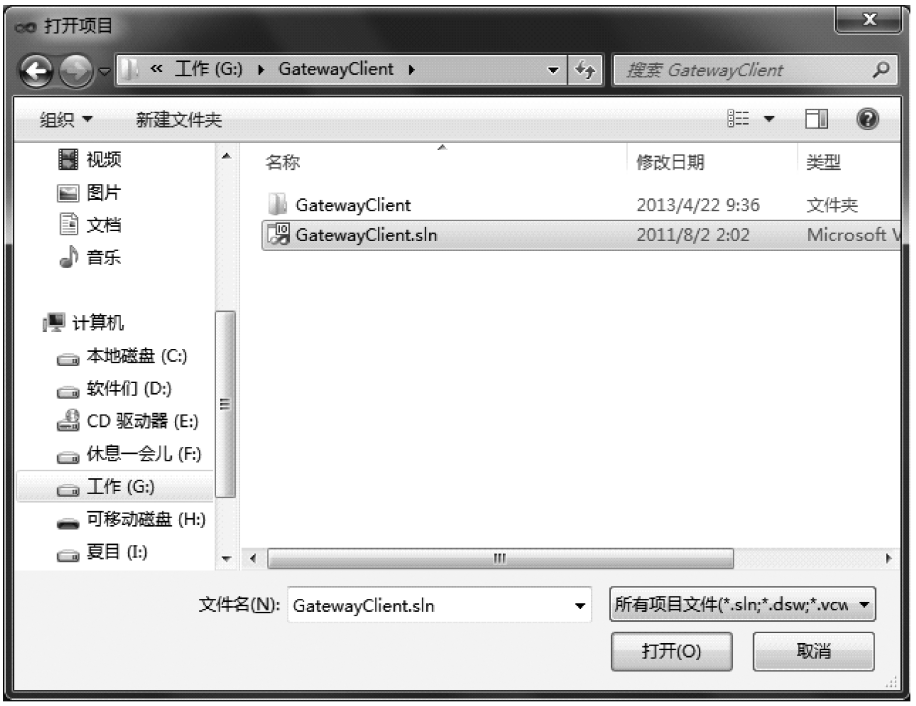


图 7-7　GatewayClient 解决方案

4）在 GatewayClient 项目中添加新建项，选择 Windows 窗体。

在如图 7-8 所示位置，单击鼠标右键选择项目“GatewayClient”，选择添加→新建项，如图 7-9 所示。

图 7-8　创建新建项

在 Visual C# 项中，选择 Windows 窗体，名称为 Form2. cs，单击添加。

5）在 GatewayClient 项目中添加新建项，选择类。

在 Visual C# 项中，选择类，名称为 say. cs，单击添加如图 7-10 所示。

6）添加完成后，开始进行编码工作。编辑 Form2. cs，如图 7-11 所示。

右键单击 Form2. cs，选择查看代码。编辑 Form2. cs 如下：

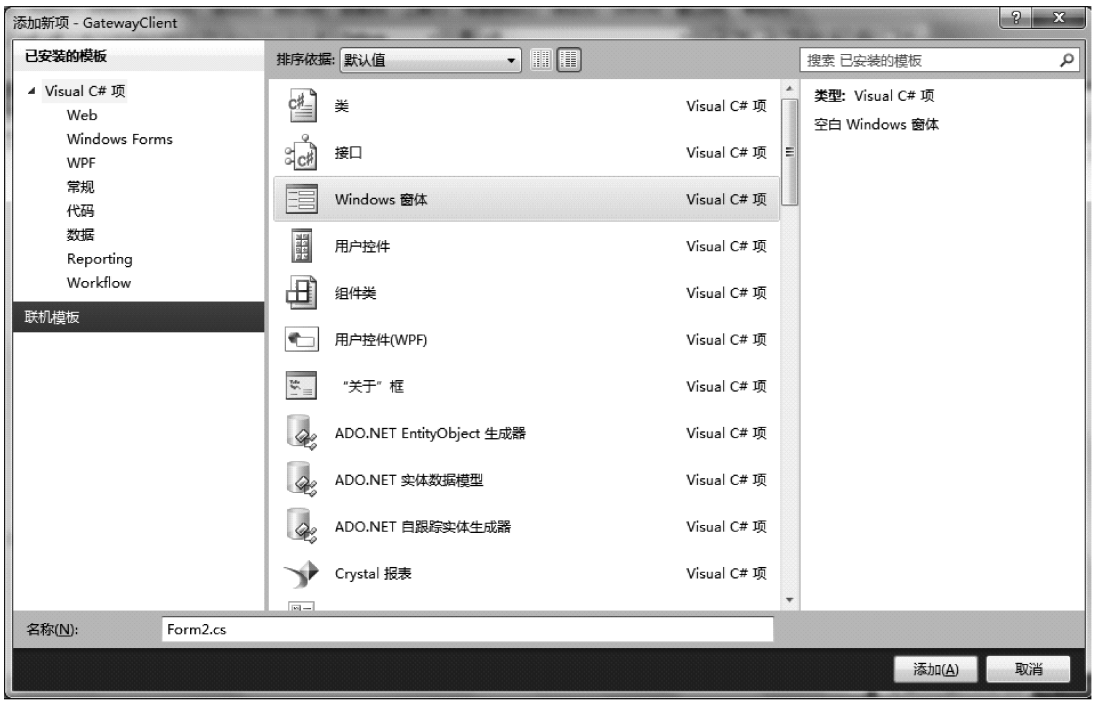


图 7-9 创建 Windows 窗体

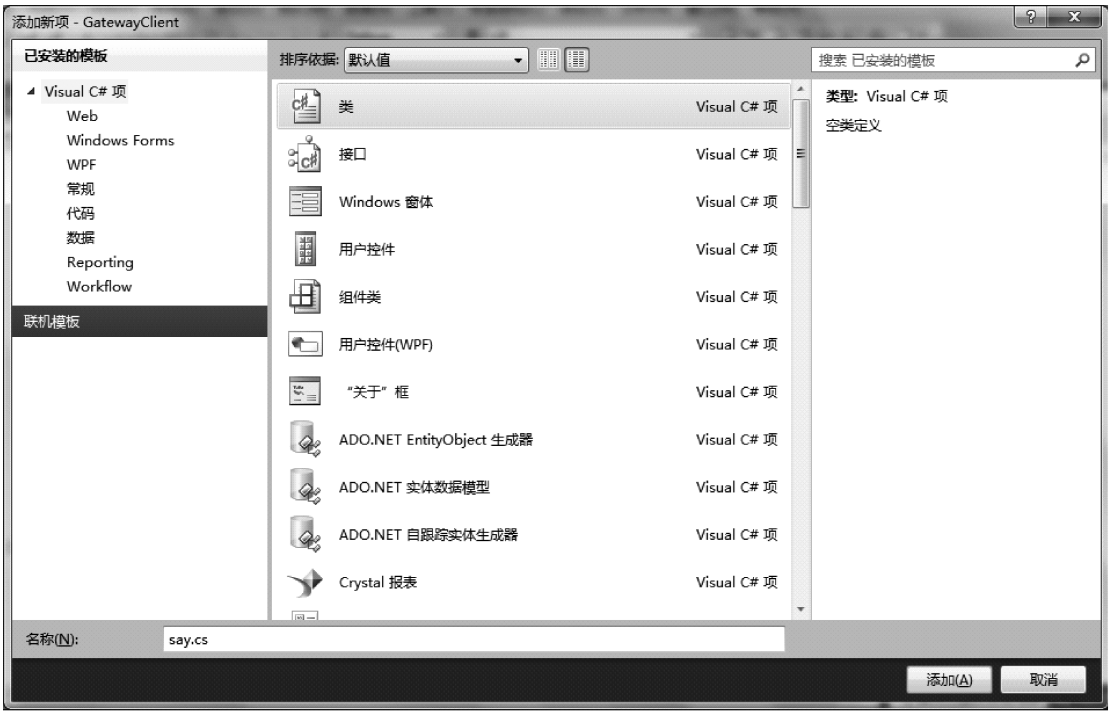


图 7-10 添加新建项

图 7-11　解决方案管理器

```
using System;
using System. Collections. Generic;
using System. ComponentModel;
using System. Data;
using System. Drawing;
using System. Linq;
using System. Text;
using System. Windows. Forms;
namespace GatewayClient
{
    public partial class Form2 : Form
    {
        TcpCli tcpClient = new TcpCli( );
        Image[ ]smile = new Image[80];
        string path = Application. StartupPath + "\\Smile\\";
        int index = 0;
        Say say = new Say( );
        public Form2( )
        {
            InitializeComponent( );
            for (int i = 100000; i < 100080; i + + )
            {
                smile[i - 100000] = Image. FromFile(path + i. ToString( ) + ". bmp");
            }
            this. timer1. Tick + = new EventHandler(timer1_Tick);
```

```
        this. timer1. Interval = 30;
        timer1. Start( );
        tcpClient. ReceivedDatagram + = new NetEvent( tcpClient_ReceivedDatagram);
        say. EventEnd + = new EndDelegate( say_EventEnd);
        say. EventStart + = new StartDelegate( say_EventStart);
        tcpClient. Connect( "115. 25. 48. 254" ,9000);
        say. SayText( "Hello. I am your housekeeper. It is my honour to serve for you" );
    }
    void say_EventStart( )
    {
        this. Show( );
        timer1. Start( );
    }
    void say_EventEnd( )
    {
        this. Hide( );
        timer1. Stop( );
    }
    private void Form2_Load( object sender, EventArgs e)
    {
        this. Hide( );
    }
    void tcpClient_ReceivedDatagram( object sender, NetEventArgs e)
    {
        say. SayText( e. Client. Datagram);
    }
    void timer1_Tick( object sender, EventArgs e)
    {
        index + + ;
        if ( index > = 80)
        {
            index = 0;
        }
        pictureBox1. Image = smile[ index];
    }
    private void Start( )
    {
        index = 0;
        this. timer1. Start( );
```

```
        }
        private void Stop( )
        {
            this. timer1. Stop( );
            index = 0;
        }
    }
}
```

7）编辑 say. cs 文件，添加语音合成功能如图 7-12 所示。

图 7-12　编辑 say. cs 文件

右键单击 say. cs，选择查看代码，编辑 say. cs，代码如下：

```
using System;
using System. Collections. Generic;
using System. Linq;
using System. Text;
using SpeechLib;
namespace GatewayClient
{
    //语音合成开始代理
    public delegate void StartDelegate( );
    //语音合成结束代理
    public delegate void EndDelegate( );
    class Say
    {
        //合成引擎
```

```
        private SpeechLib. SpVoice_vo = new SpeechLib. SpVoice( );
        //异步合成方式
        private SpeechLib. SpeechVoiceSpeakFlags_SSF =
SpeechVoiceSpeakFlags. SVSFlagsAsync;
        //所有的角色列表
        private List < string > charactorList = null;
        //我们选择的语音合成角色
        private string charactor = " Microsoft Lili" ;
        //语音合成开始事件
        public event StartDelegate EventStart;
        //语音合成结束事件
        public event EndDelegate EventEnd;
        public Say( )
        {
            _vo. EndStream + = new
_ISpeechVoiceEvents_EndStreamEventHandler( _vo_EndStream) ;
            _vo. StartStream + = new
_SpeechVoiceEvents_StartStreamEventHandler( _vo_StartStream) ;
        }

        void_vo_StartStream( int StreamNumber, object StreamPosition)
        {
            if ( EventStart ! = null)
            {
                EventStart( ) ;
            }
        }
        public string Charactor
        {
            get { return charactor; }
            set { charactor = value; }
        }
        public List < string > Charactors
        {
            get
            {
                if ( charactorList = = null)
                {
                    GetCharactors( ) ;
```

```
            }
            return charactorList;
        }
    }
    private void GetCharactors()
    {
        charactorList = new List < string > ();
        SpeechLib. ISpeechObjectTokens voices = _vo. GetVoices();
        foreach (SpeechLib. ISpeechObjectToken t in voices)
        {
            charactorList. Add(t. GetAttribute("Name"). ToString());
        }
    }
    void_vo_EndStream(int StreamNumber,object StreamPosition)
    {
        if (EventEnd! = null)
        {
            EventEnd();
        }
    }
    public bool SayText(string charactor,string text)
    {
        if (charactor = = null)
        {
            return false;
        }
        _vo. Speak(string. Empty,SpeechVoiceSpeakFlags. SVSFPurgeBeforeSpeak);
        _vo. Voice = _vo. GetVoices("name = " + charactor,""). Item(0);
        _vo. Speak(text,_SSF);
        return true;
    }
    public bool SayText(string text)
    {
        if (charactor = = null)
        {
            return false;
        }
        _vo. Voice = _vo. GetVoices("name = " + charactor,""). Item(0);
        _vo. Speak(text,_SSF);
```

```
            return true;
        }
    }
}
```

8）生成解决方案如图 7-13 所示。

右键单击选择项目 GatewayClient，选择生成。

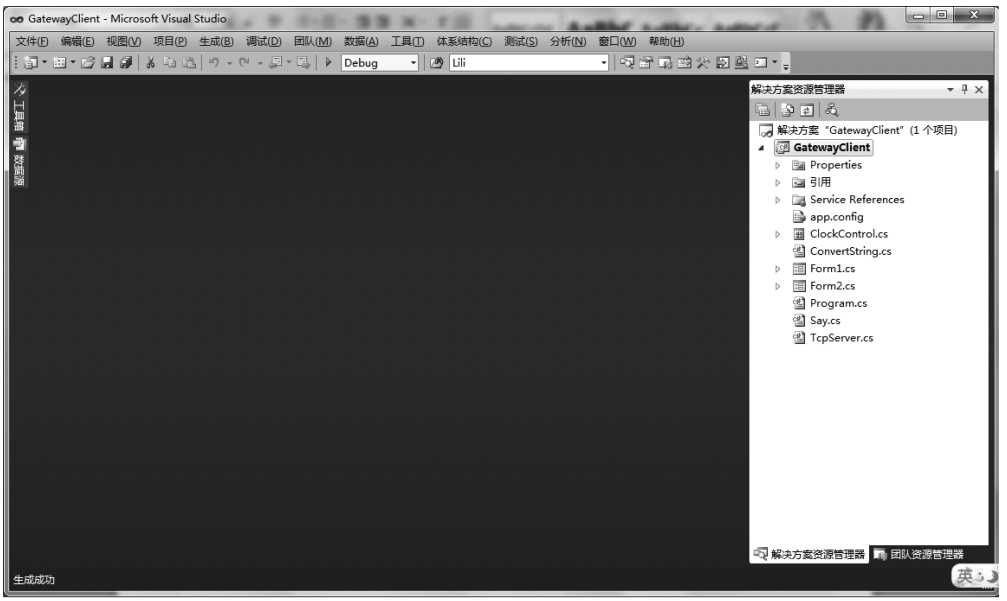


图 7-13　生成解决方案

若程序编写无误，IDE 左下角将提示“生成成功”。

9）启动调试。

单击 Debug 左边的绿色三角标志（如

），开启程序调试运行过程，程序启动成功后，将出现如图 7-14 所示的虚拟管家，伴随动画，播放语音，“您好，我是您的管家小丽，很高兴为您服务。”

虚拟管家消失后，单击 Form1 中的连接网关如图 7-15 所示。

连接网关成功后，出现虚拟管家，伴随动画播放语音，“您好，客人，欢迎来到智能空间实验室如图 7-16 所示。”

虚拟管家消失后，便可通过网关客户端对智能空间实验室中的设备进行操作。

如在网关中选择“客厅主灯”，如图 7-17 所示。

切换到设备操作选项卡，如图 7-18 所示。

在操作列表中可以选择 Open 和 Close 操作，对应开关客厅主灯。选择 Open 操作，单击“调用”，如图 7-19 所示。

出现虚拟管家动画，伴随语音，“客人请求了客厅主灯的打开。主灯操作。”，如图 7-20 所示。

图 7-14　欢迎界面

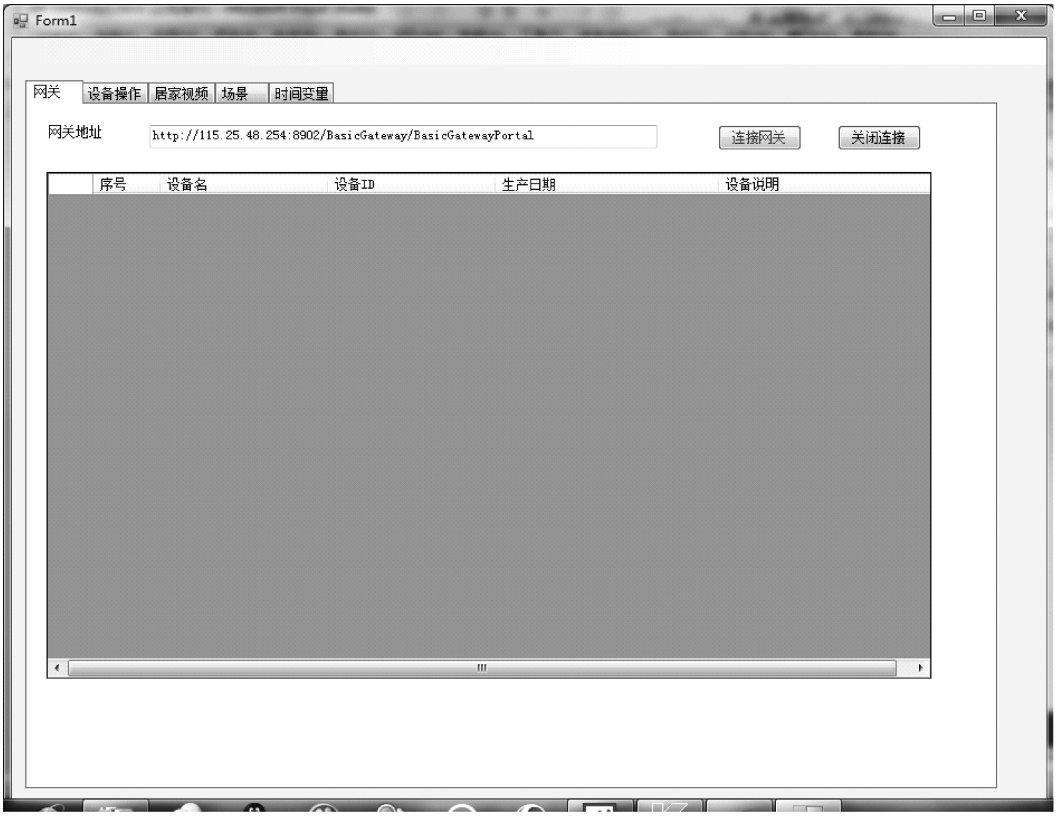


图 7-15　连接网关界面

图 7-16 欢迎界面

	序号	设备名	设备ID	生产日期	设备说明
	0	物联网空调	AirCondition	2011/4/5 0:00:00	这是一个空调
	1	湿度传感器	HumidityTransducer	2011/4/5 0:00:00	这是一个湿度传感器
	2	光照传感器	LightSensor01	2011/4/5 0:00:00	这是一个光照传感器
	3	光照传感器	LightSensor02	2011/4/5 0:00:00	这是一个光照传感器
▶	4	客厅主灯	MainLamp	2011/4/5 0:00:00	这是一个红外灯
	5	卫星灯	SubLamps	2011/4/5 0:00:00	这是一个红外灯
	6	温度传感器	TemperatureTransducer01	2011/4/5 0:00:00	这是一个温度传感器
	7	温度传感器	TemperatureTransducer02	2011/4/5 0:00:00	这是一个温度传感器
	8	温度传感器	TemperatureTransducer03	2011/4/5 0:00:00	这是一个温度传感器
	9	健康电视机	TV	2011/4/5 0:00:00	这是一个电视机
	10	卧室辅灯	ViceLamp	2011/4/5 0:00:00	这是一个红外灯
	11	网络摄像头	WebCamera1	2011/4/5 0:00:00	这是一个网络摄像头
	12	网络摄像头	WebCamera2	2011/4/5 0:00:00	这是一个网络摄像头
	13	网络摄像头	WebCamera3	2011/4/5 0:00:00	这是一个网络摄像头
	14	网络摄像头	WebCamera4	2011/4/5 0:00:00	这是一个网络摄像头
	15	机器人	Robot	2011/4/5 0:00:00	这是一个机器人
	16	空气清新机	AirCleaner	2011/4/5 0:00:00	这是一个空气清新机

图 7-17 选择设备

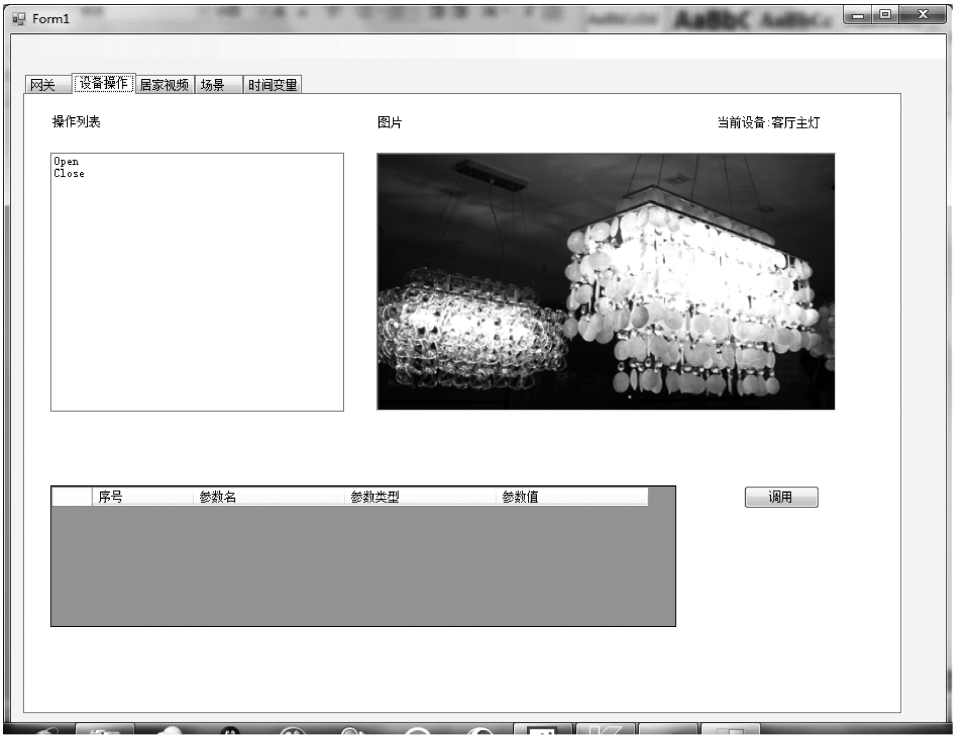


图 7-18　执行命令

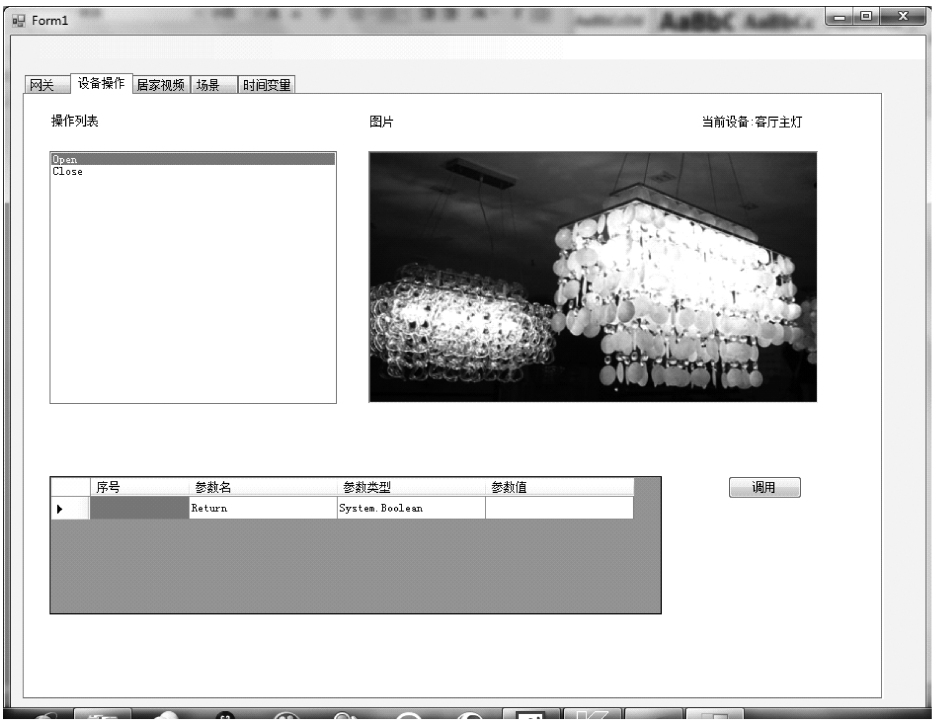


图 7-19　操作结果演示

图 7-20　语音提示

选择 Close 操作，单击“调用”，如图 7-21 所示。

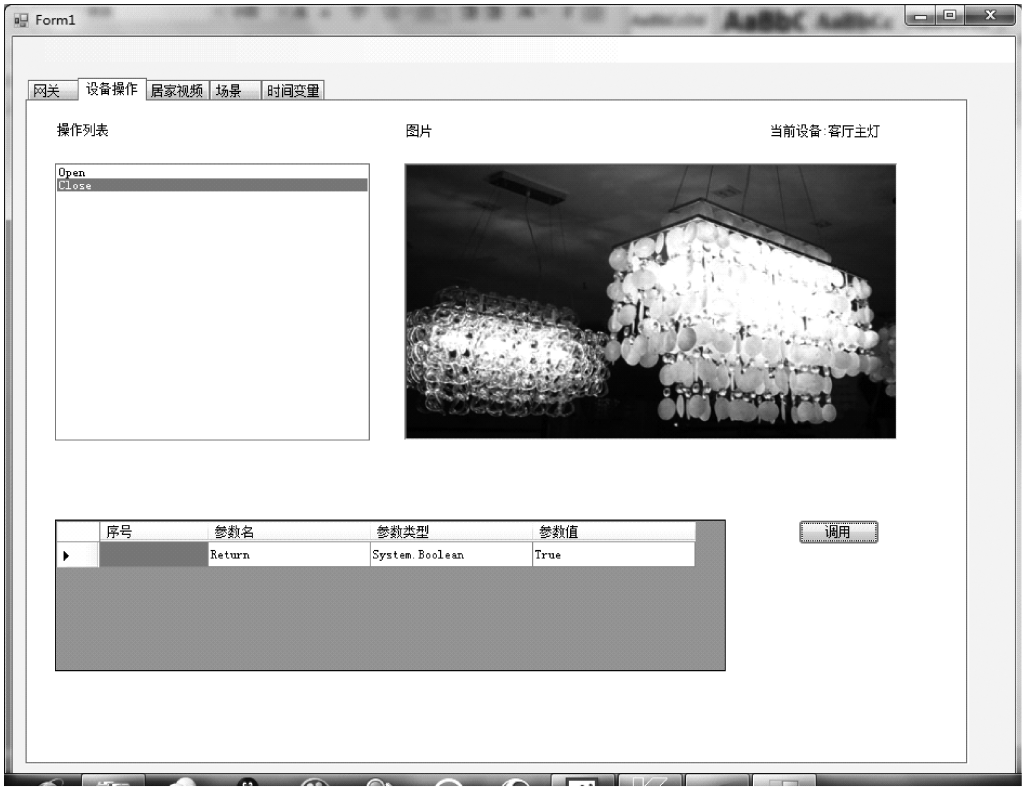


图 7-21　操作结果演示

出现虚拟管家动画，伴随语音，“客人请求了客厅主灯的关闭。主灯操作。”，如图 7-22 所示。

图 7-22　语音提示

同理可以对网关列表中的任一设备进行操作。

但请注意务必确保所有设备最后进行的都是 Close 操作，因为通过网关连接将真实地开启或关闭设备。节约能源，从你我做起。

测试结束后，单击“关闭连接”，如图 7-23 所示。

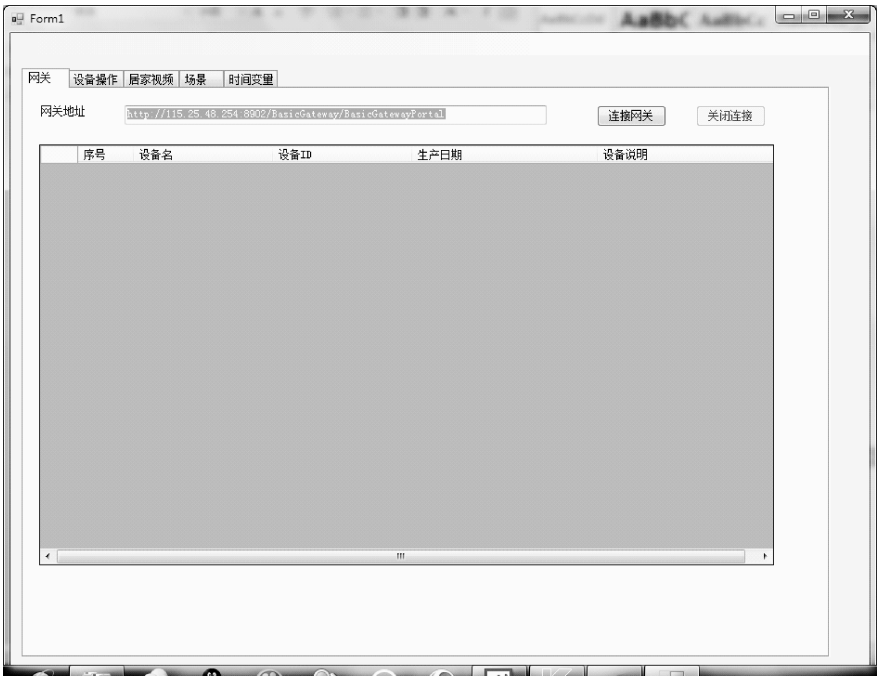


图 7-23　关闭连接

7.6 思考题

1）软件调试中遇到的问题都有哪些？分别是怎样解决的？

2）除了本实验以外，请改变虚拟管家的语音提示内容。

7.7 实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤（包括设计方案、软件流程图、硬件原理图等）以及调试过程、心得体会，最后要附上程序，并按要求回答思考题1）、2）。

参 考 文 献

[1] 王志良，李正熙，解仑. 信息社会中的自动化新技术［M］. 北京：机械工业出版社. 2004.

[2] Bustamante，M L. 学习 WCF 从原理到实践［M］. 庞引明，侯伯薇，译. 北京：电子工业出版社. 2008.

[3] 董士海. 人机交互的进展及面临的挑战［J］. 计算机辅助设计与图形学学报，2004（1）.

[4] 王志良. 人工心理学——关于更接近人脑工作模式的科学［J］. 北京科技大学学报，2000（10）.

[5] 袁保宗. 新一代（第四代）人机交互的概念框架特征及关键技术［J］. 电子学报，2003（12）.

[6] J Johnston，S M Fiore，C Paris，et al. Application of cognitive load theory to developing a measure of team decision efficiency［J］. Military Psychology，2006.

[7] K Dautenhahn，M Walters，S Woods，et al. How may I serve you? A robot companion approaching a seated person in a helping context［J］. Salt Lake City：in ACM Conference on Human-Robot Interaction（HRI），2006.

[8] 张全，朱珍民. 泛在设备互联桥接器设计及实现［J］. 计算机工程，2011（20）.

[9] 张宝鹏，史元春. 智能空间中高效率的服务合成策略［J］. 清华大学学报，2008（7）.

[10] 谢伟凯. 智能空间关键支撑技术的研究［D］. 北京：清华大学，2003.

第 8 章　移动终端之安卓开发实验

8.1　实验目的

1）了解使用安卓手机进行远程监控的方法原理。
2）掌握安卓手机应用程序的基本开发流程。
3）熟悉安卓应用程序开发的集成环境 Eclipse。

8.2　实验内容

在 Linux 环境下，搭建 Eclipse + ADT 系统，实现安卓手机远程监控功能，达到掌握原理、熟悉应用的效果。

8.3　实验所用仪表及设备

1）硬件：PC、安卓手机、网络摄像头各一。
2）软件：Ubuntu12.04 操作系统、安卓 4.0 以上、Eclipse IDE、Android SDK。

8.4　实验原理

8.4.1　系统简介

安卓（见图 8-1）是 Google 在 2007 年 11 月 5 日公布的基于 Linux 平台的开源智能手机操作系统名称。该平台由操作系统、中间件、用户界面和应用软件组成。

图 8-1　安卓 logo

8.4.2　发展历程及发行版本

2003 年 10 月，Andy Rubin 等人创建安卓公司，并组建安卓团队。

2005 年 8 月 17 日，Google 低调收购了成立仅 22 个月的高科技企业安卓及其团队。安迪鲁宾成为 Google 公司工程部副总裁，继续负责安卓项目。

2007 年 11 月 5 日，Google 正式向外界展示了这款名为安卓的操作系统，并且在这天 Google 宣布建立一个全球性的联盟组织，该组织由 34 家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成，并与 84 家硬件制造商、软件开发商及电信营运商组成开放

手持设备联盟（Open Handset Alliance）来共同研发改良安卓系统，这一联盟将支持谷歌发布的手机操作系统以及应用软件，Google 以 Apache 免费开源许可证的授权方式，发布了安卓的源代码。

2008 年，在 GoogleI/O 大会上，Google 提出了 AndroidHAL 架构图 8，在同年 8 月 18 号，安卓获得了美国联邦通信委员会（FCC）的批准，在 2008 年 9 月，Google 正式发布了安卓 1.0 系统，这也是安卓系统最早的版本。

2009 年 4 月，Google 正式推出了安卓 1.5 这款手机，从安卓 1.5 版本开始，Google 开始将安卓的版本以甜品的名字命名，安卓 1.5 命名为 Cupcake（纸杯蛋糕）。该系统与安卓 1.0 相比有了很大的改进。

2009 年 9 月，Google 发布了安卓 1.6 的正式版，并且推出了搭载安卓 1.6 正式版的手机 HTC Hero（G3）。安卓 1.6 也有一个有趣的甜品名称，它被称为 Donut（甜甜圈）。

2010 年 2 月，Linux 内核开发者 Greg Kroah-Hartman 将安卓的驱动程序从 Linux 内核“状态树”（“staging tree”）上除去，从此，安卓与 Linux 开发主流将分道扬镳。在同年 5 月，Google 正式发布了安卓 2.2 操作系统。Google 将安卓 2.2 操作系统命名为 Froyo，翻译完名为冻酸奶。

2010 年 10 月，Google 宣布安卓系统达到了第一个里程碑，即电子市场上获得官方数字认证的安卓应用数量已经达到了 10 万个，安卓系统的应用增长非常迅速。在 2010 年 12 月，Google 正式发布了安卓 2.3 操作系统 Gingerbread（姜饼）。

2011 年 1 月，Google 称每日的安卓设备新用户数量达到了 30 万部，到 2011 年 7 月，这个数字增长到 55 万部，而安卓系统设备的用户总数达到了 1.35 亿。

2011 年 8 月 2 日，安卓手机已占据全球智能机市场 48% 的份额，并在亚太地区市场占据统治地位，终结了 Symbian（塞班系统）的霸主地位，跃居全球第一。

2011 年 9 月，安卓系统的应用数目已经达到了 48 万，而在智能手机市场，安卓系统的占有率已经达到了 43%。继续排在移动操作系统首位。Google 发布全新的安卓 4.0 操作系统，这款系统被 Google 命名为 Ice Cream Sandwich（冰激凌三明治）。

2012 年 1 月 6 日，Google Android Market 已有 10 万开发者推出超过 40 万活跃的应用，大多数的应用程序为免费。Android Market 应用程序商店目录在新年首周周末突破 40 万基准，距离突破 30 万应用仅 4 个月。在 2011 年早些时候，Android Market 从 20 万增加到 30 万应用也花了 4 个月。

8.4.3 系统架构

安卓的系统架构和其操作系统一样，采用了分层的架构。如图 8-2 所示，安卓分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 内核层。

1. 应用程序

安卓会同一系列核心应用程序包一起发布，该应用程序包包括客户端、SMS（短消息）程序、日历、地图、浏览器、联系人管理程序等。所有的应用程序都是使用 Java 语言编写的。

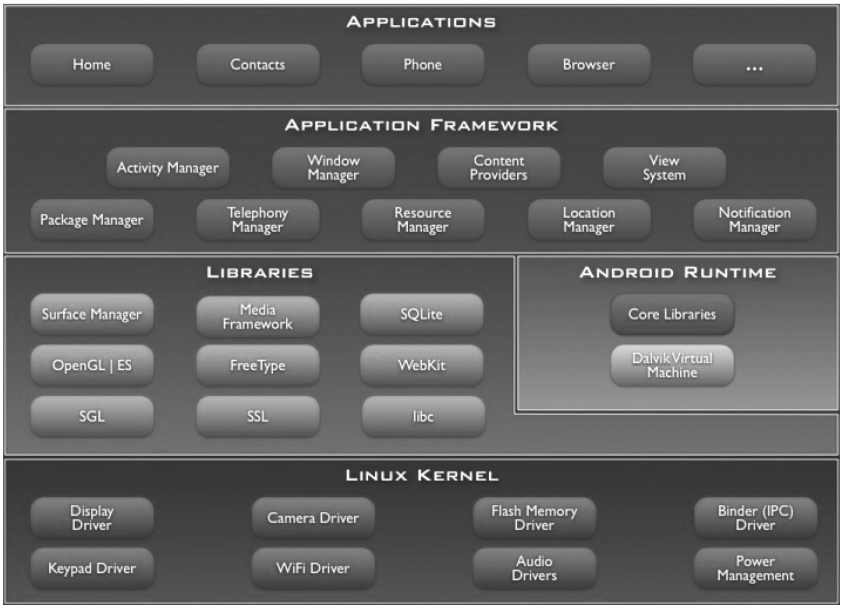


图 8-2　安卓架构

2. 应用程序框架

开发人员也可以完全访问核心应用程序所使用的 API 框架。该应用程序的架构设计简化了组件的重用；任何一个应用程序都可以发布它的功能块，并且任何其他的应用程序都可以使用其所发布的功能块（不过要遵循框架的安全性）。同样，该应用程序重用机制也使用户可以方便地替换程序组件。

隐藏在每个应用后面的是一系列的服务和系统，其中包括：丰富而又可扩展的视图（Views），可以用来构建应用程序，它包括列表（Lists）、网格（Grids）、文本框（Text boxes）、按钮（Buttons），以及可嵌入的 Web 浏览器。

内容提供器（Content Providers）使得应用程序可以访问另一个应用程序的数据（如联系人数据库），或者共享它们自己的数据。

资源管理器（Resource Manager）提供非代码资源的访问，如本地字符串、图形和布局文件（Layout files）。

通知管理器（Notification Manager）使得应用程序可以在状态栏中显示自定义的提示信息。

活动管理器（Activity Manager）用来管理应用程序生命周期并提供常用的导航回退功能。

3. 系统运行库

安卓包含一些 C/C++库，这些库能被安卓系统中不同的组件使用。它们通过安卓应用程序框架为开发者提供服务。以下是一些核心库：

系统 C 库：一个从 BSD 继承来的标准 C 系统函数库 Libc，它是专门为基于 Embedded Linux 的设备定制的。

媒体库：基于 PacketVideo OpenCORE；该库支持多种常用的音频、视频格式回放和录

制，同时支持静态图像文件。编码格式包括 MPEG4、H.264、MP3、AAC、AMR、JPG 和 PNG。

Surface Manager：对显示子系统的管理，并且为多个应用程序提供了 2D 和 3D 图层的无缝融合。

LibWebCore：一个最新的 Web 浏览器引擎用，支持安卓浏览器和一个可嵌入的 Web 视图。

8.4.4 平台优势

1. 开放性

在优势方面，安卓平台首先就是其开发性，开发的平台允许任何移动终端厂商加入到安卓联盟中来。显著的开放性可以使其拥有更多的开发者，随着用户和应用的日益丰富，一个崭新的平台也将很快走向成熟。

开放性对于安卓的发展而言，有利于积累人气，这里的人气包括消费者和厂商，而对于消费者来讲，最大的受益正是丰富的软件资源。开放的平台也会带来更大竞争，如此一来，消费者将可以用更低的价位购得心仪的手机。

2. 不受束缚

在过去很长的一段时间，特别是在欧美地区，手机应用往往受到运营商制约，使用什么功能接入什么网络，几乎都受到运营商的控制。自从 2007 年 iPhone 上市后，用户可以更加方便地连接网络，运营商的制约减少了。随着 EDGE、HSDPA 这些 2G 至 3G 移动网络的逐步过渡和提升，手机随意接入网络已不是运营商口中的笑谈。

3. 丰富的硬件

这一点还是与安卓平台的开放性相关，由于安卓的开放性，众多的厂商会推出千奇百怪、功能特色各具的多种产品。功能上的差异和特色，却不会影响到数据同步、甚至软件的兼容，如同从诺基亚 Symbian 风格手机一下改用苹果 iPhone，同时还可将 Symbian 中优秀的软件带到 iPhone 上使用、联系人等资料更是可以方便地转移。

4. 方便开发

安卓平台提供给第三方开发商一个十分宽泛、自由的环境，不会受到各种条条框框的限制，可想而知，会有多少新颖别致的软件会诞生。但也有其两面性，血腥、暴力、情色方面的程序和游戏如何控制正是留给安卓的难题之一。

8.5 实验程序设计

8.5.1 程序执行流程

如图 8-3 所示，本程序需要实现获取远程摄像机实况画面的功能。当程序启动之后，等待用户选择操作。当用户选择开始监控后，将发起 HTTP 连接请求到远程网络摄像机，取得摄像机当前的监控画面后，将其显示在屏幕上。如果用户不选择停止监控，将自动不断向摄像机请求新画面并刷新。当用户选择停止监控时，将会断开到远程摄像机的 HTTP 连接，回到程序等待界面，直到用户选择再次开启监控或者程序退出。

8.5.2　开发环境配置

系统软件运行于安卓设备之上，使用 Java 语言编写。选用较流行的 Eclipse IDE 进行开发。Eclipse 软件可运行于 Windows、Linux 和 Mac OS 之上，在此以 Linux 操作系统中的 Ubuntu 发行版为例，进行开发环境的搭建。

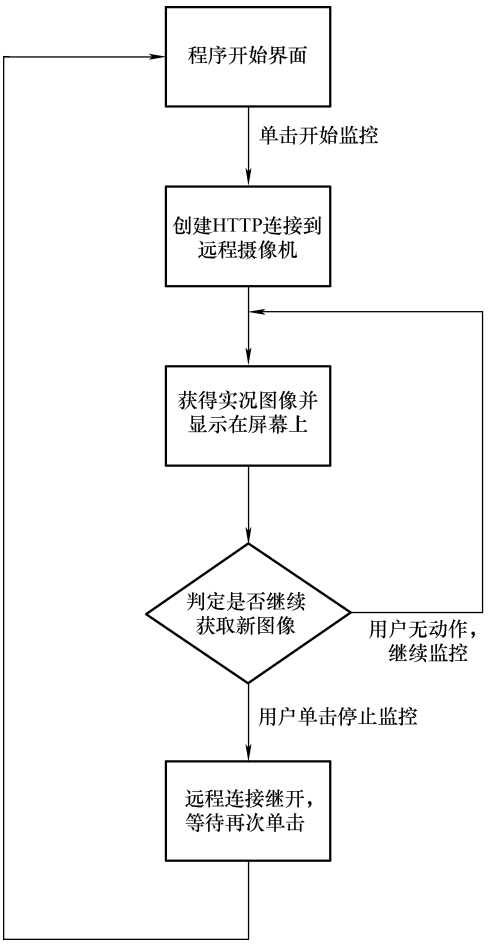


图 8-3　程序执行流程

1. Java 开发环境配置

Eclipse 开发环境本身使用 Java 语言编写，需要底层的 Java 运行环境（JRE）来运行，因此首先要配置 Java 开发环境。

确定系统连接到 Internet，打开终端窗口，如图 8-4 所示，输入：“sudo apt-get install openjdk-6-jre”，即可在线自动安装 Java 开发环境（版本为 openjdk-6）。

等待安装过程完毕即可。

检查 Java 开发环境是否配置好，在终端输入：“java-version”，显示版本信息如图 8-5 所示。

2. Eclipse IDE 的安装与配置

访问 http://developer.android.com/sdk/index.html，下载最新版 Eclipse + ADT 开发环境。将下载后的压缩包解压到同一文件夹下，两个子文件夹如图 8-6 所示，即为已经配置好的Eclipse开发环境和开发安卓程序的插件。

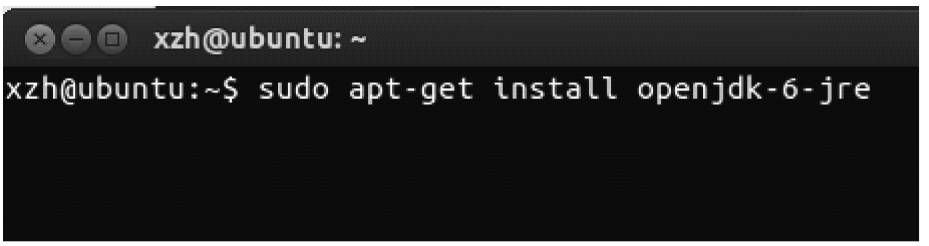


图 8-4　安装 Java 开发环境

在终端中输入：“sudo gedit /.bashrc”，打开此文件，写入一行：“export PATH =/home/xzh/java/sdk/platform-tools/：$ PATH.”，保存退出，其中/home/xzh/java/要替换为自己系统中 sdk 文件夹所在的目录。

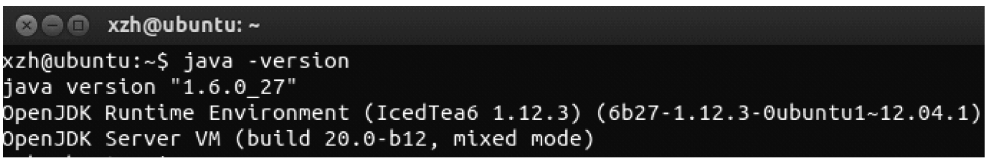


图 8-5 查看 Java 版本信息

进入 eclipse 文件夹，打开 eclipse 执行文件，即可使用 eclipse 环境开发安卓应用程序。

3. 使用 Eclipse + ADT 新建一个安卓工程

图 8-6 Eclipse + ADT 开发环境

打开 Eclipse 开发环境，在 File→New→Project 中，选择 Android Application Project，单击“Next”继续，如图 8-7 所示。

填写应用程序（Application）的名称、工程名称和软件包的名称，其中 Application Name（中英文均可）即为安装在手机上后应用程序的显示名称。

然后选择应用程序最低支持的安卓版本，开发所针对的安卓版本，以及编译程序所使用的安卓版本，如图 8-8 所示，我们对工程命名为 Test_camera，开发版本为安卓 4.1。

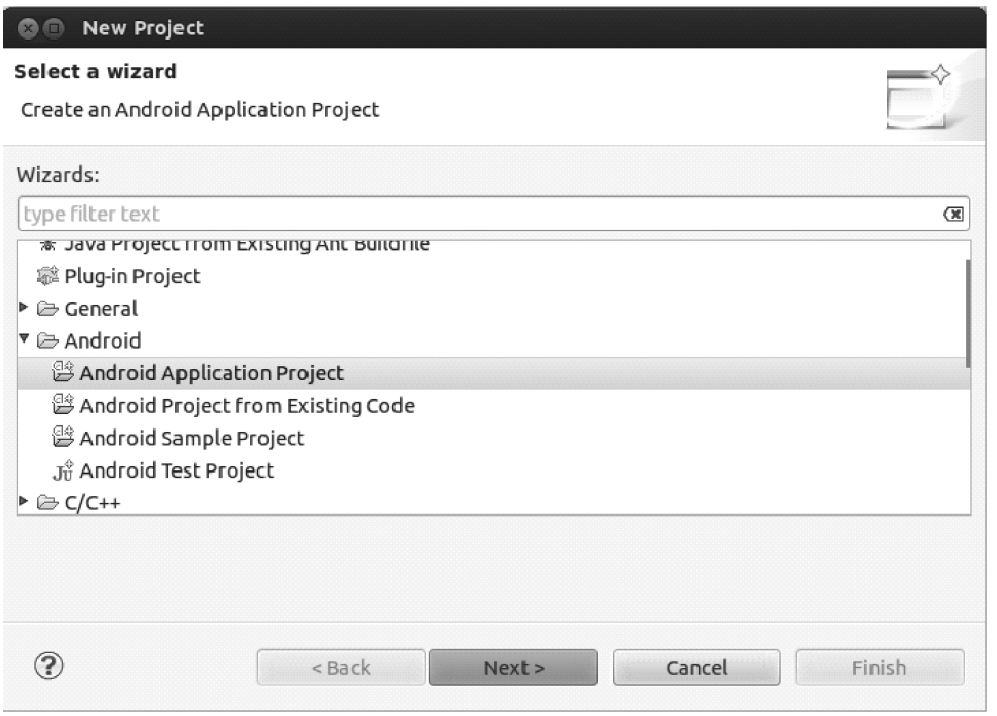


图 8-7 开发 Android Project

如无特殊设置，保持默认配置即可，单击“Next”继续，如图 8-9 所示。

选择应用程序的图标，如图 8-10 所示，单击“Browse”可替换为自定义图标，单击

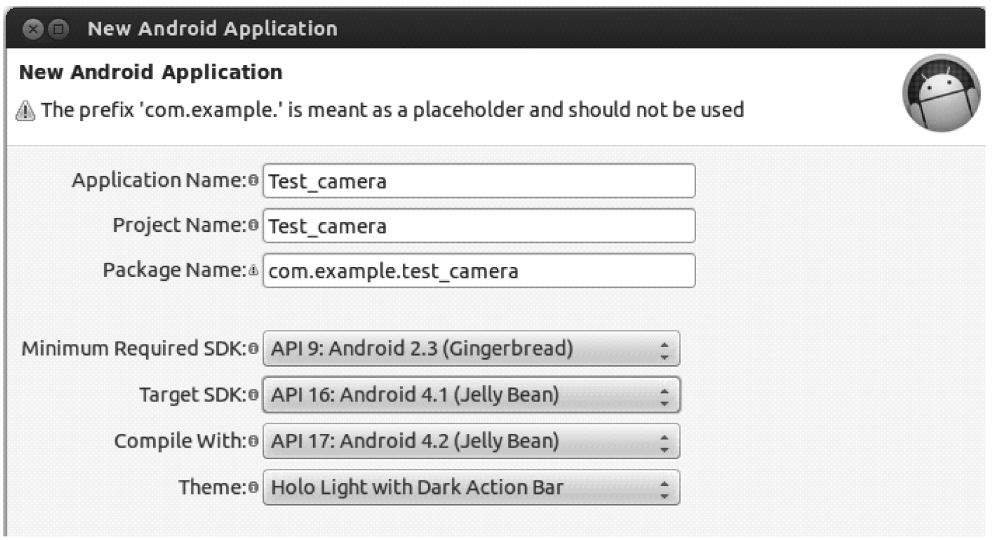


图 8-8　设置应用程序

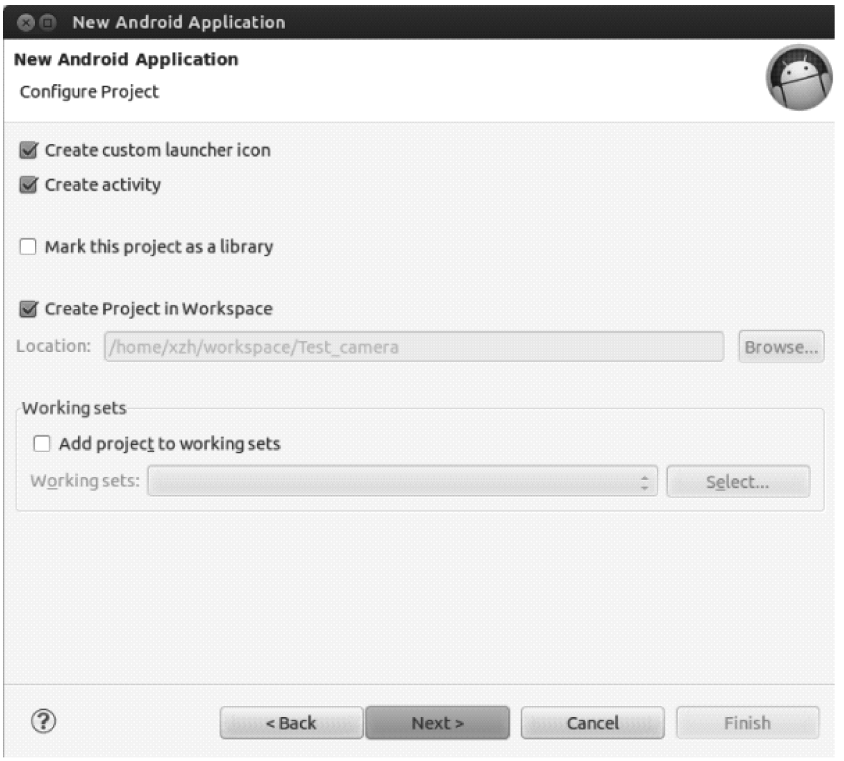


图 8-9　开发 Android Project

“Next”继续。

创建一个 Activity，即应用程序的界面，使用 Blank Activity，即空白的界面，以便于自己配置布局，如图 8-11 所示。

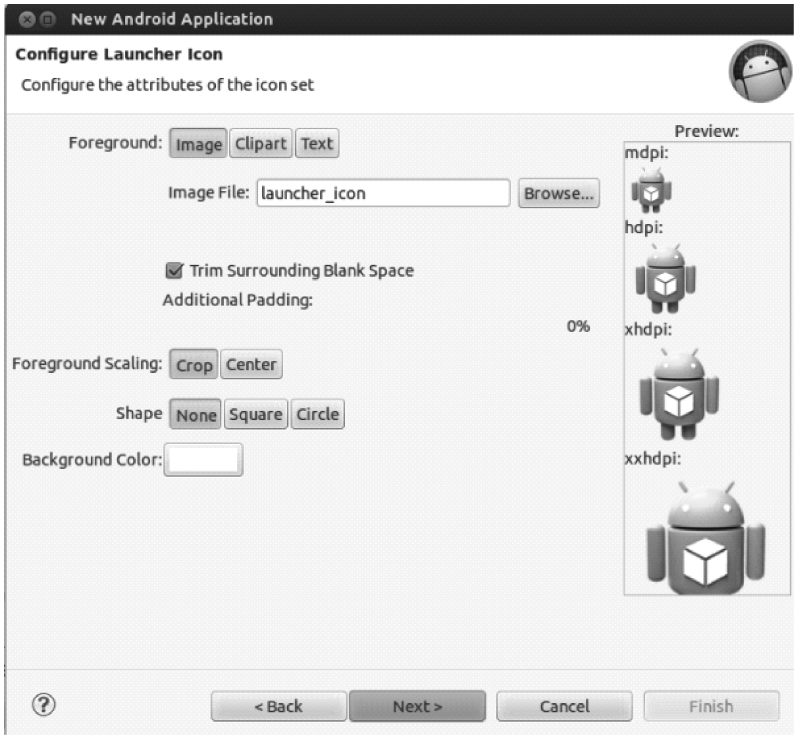


图 8-10　选择应用图标

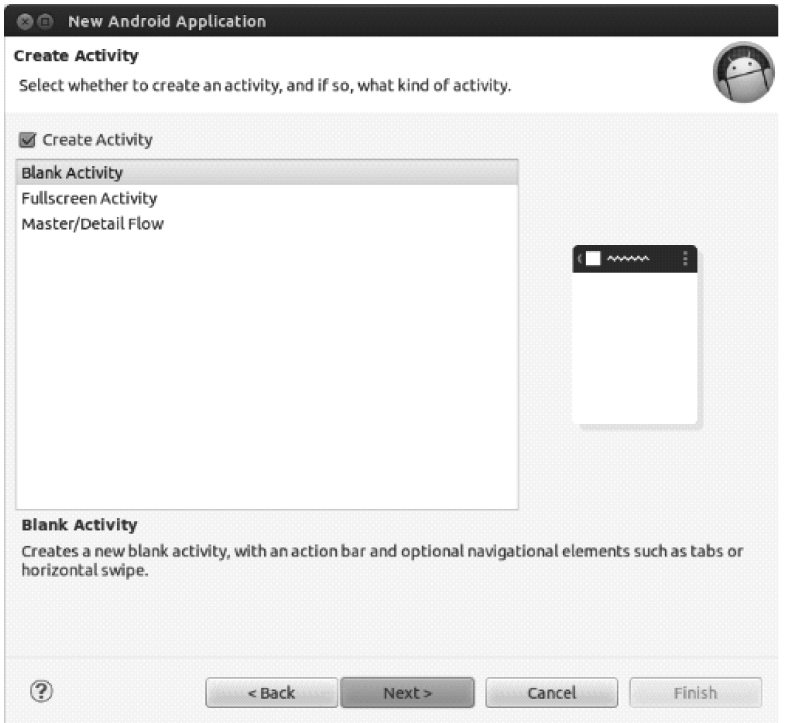


图 8-11　新建一个 Activity

填写 Activity 以及它的布局文件（Layout）的名称，单击“Finish”，即完成了新建的这个工程（Project）的首个界面的设置，如图 8-12 所示。

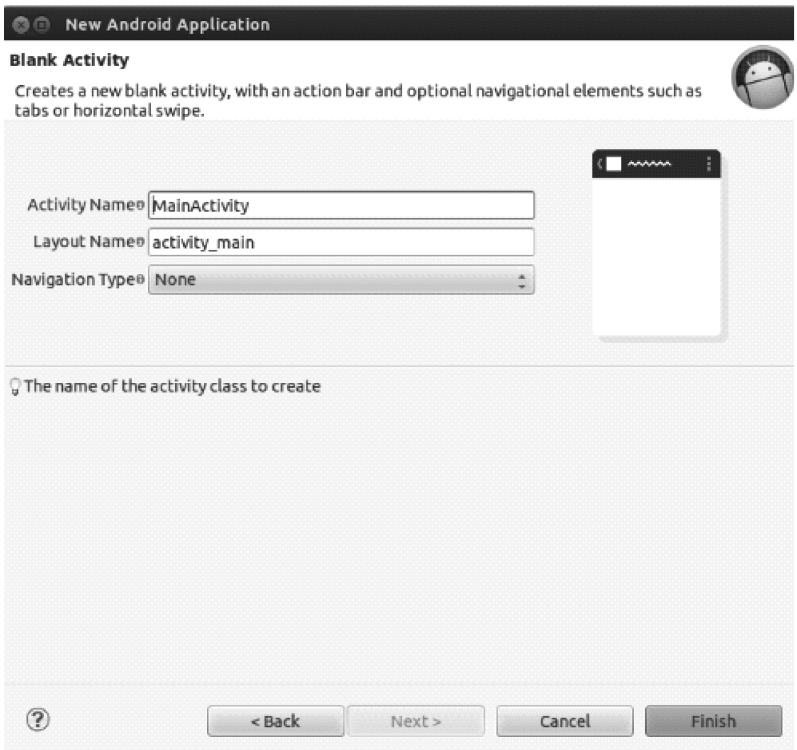


图 8-12　完成设置

8.5.3　设置应用程序的界面布局 Layout

如图 8-13 所示，上一步完成设置后，MainActivity 的布局文件（Layout）已经自动打

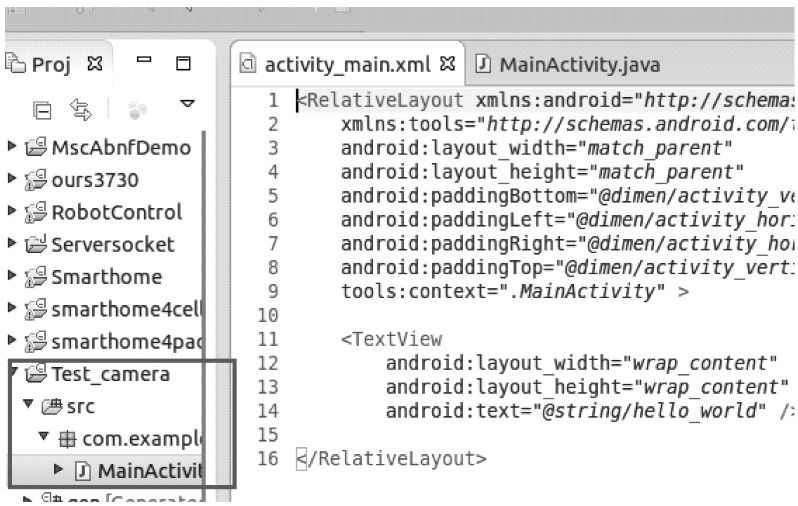


图 8-13　开始编辑工程

开。如单击关闭了，在 Eclipse 左侧工程目录 Test_camera 下，找到 res/layout 文件夹即可找到（见图 8-14）。res 文件夹存放工程所需的图片、布局文件等资源。

在 Layout 文件的窗口中，单击下方的“Graphical Layout”，即可观看当前已保存的布局视图。在编辑过程中可随时在此查看布局效果。如图 8-15 所示，安卓工程建立之后，系统会默认显示一个“Hello world!”欢迎字样。

单击图 8-13 中右侧的“activity_main. xml”，回到代码视图，如图 8-16 所示，选中系统自带的 TextView 部分，此处即为显示 Hello world! 字样的组件，删除后保存文件，即不再显示，实现修改默认布局组件。

Relative Layout 为安卓应用程序界面的布局方式中灵活的一种，我们可以在其中指定图形组件的多种排列方式。为了显示监控画面，在此需要在主 Layout 中嵌套一个布局，如图 8-17 所示。

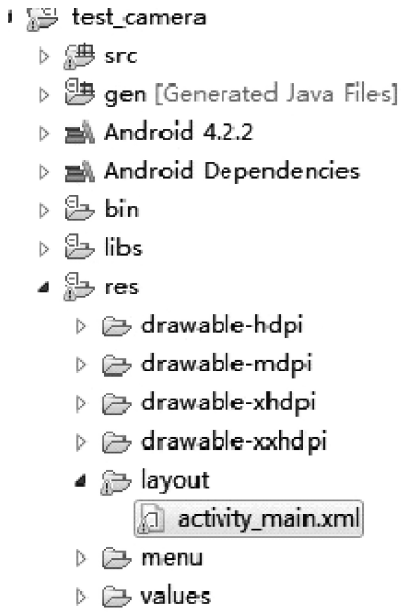


图 8-14 res/layout 文件夹

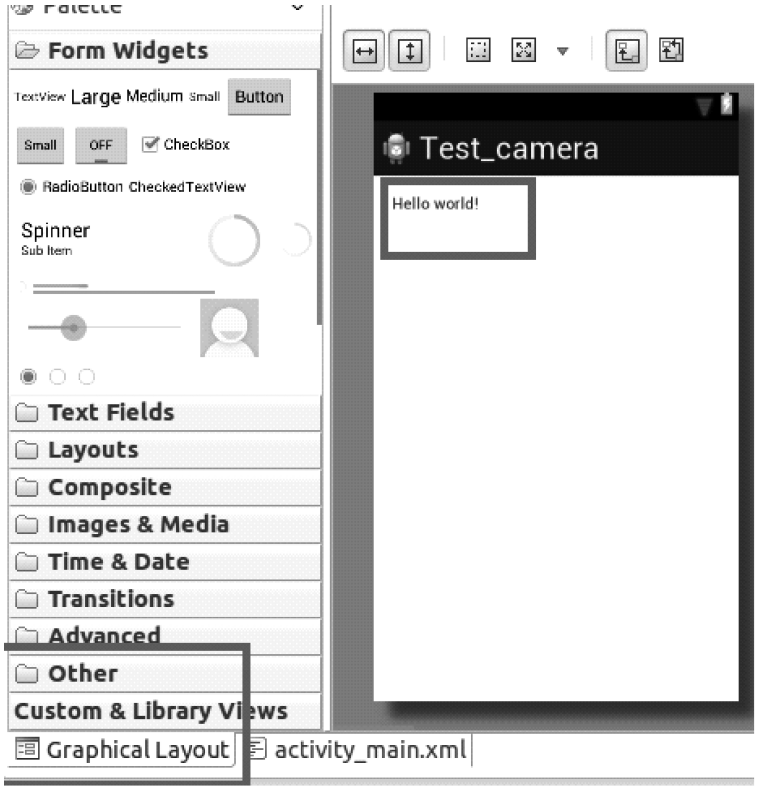


图 8-15 编辑布局文件

```
10
11      <TextView
12          android:layout_width="wrap_content"
13          android:layout_height="wrap_content"
14          android:text="@string/hello_world" />
15
```

图 8-16　修改默认布局组件

```
*activity_main.xml   MainActiv
 1  <RelativeLayout xmlns:androi
 2      xmlns:tools="http://sche
 3      android:layout_width="ma
 4      android:layout_height="m
 5      android:paddingBottom="@
 6      android:paddingLeft="@di
 7      android:paddingRight="@d
 8      android:paddingTop="@dim
 9      tools:context=".MainActi
10
11      <RelativeLayout
12          >
13
14      </RelativeLayout>
15
16  </RelativeLayout>
```

图 8-17　嵌套布局

在内嵌的布局中，对 RelativeLayout 组件做如下参数设置，自定义监控画面组件属性。

```
< RelativeLayout
    android:id = "@+id/inner_layout"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_alignParentTop = "true"
    android:layout_centerHorizontal = "true"
    >
    < ImageView
        android:id = "@+id/camera_view"
        android:layout_width = "320dip"
        android:layout_height = "240dip"
        android:longClickable = "true"/ >
</RelativeLayout >
```

组件的参数定义，每行单列一项，都以“android:”开头声明。其中 id 表示对组件的命名，每个应用程序中不可重复，以便应用程序调用组件显示在界面上；layout_width 和 height 分别为组件的宽、高；wrap_content 表示设置组件宽高为包裹其内部子组件的大小为准；

alignParentTop 表示此组件对其他的父控件（即包裹它的外层组件）的顶部；centerHorizontal 表示此组件在界面显示时水平居中。

接下来要在此内嵌的布局里添加用来显示监控画面的子组件，此类组件必须声明为 < ImageView > 。同上，设置组件 id。并且将显示画面的宽高设置为固定值，其中 dip 表示与硬件设备显示屏的像素点数有关；longClickable = “true” 表示此组件允许用户长时间点击，以便于后续开发触控拖曳操作。

此时再切换到图形视图，已经可以看到刚刚定义的显示监控画面的窗口了，如图 8-18 所示。

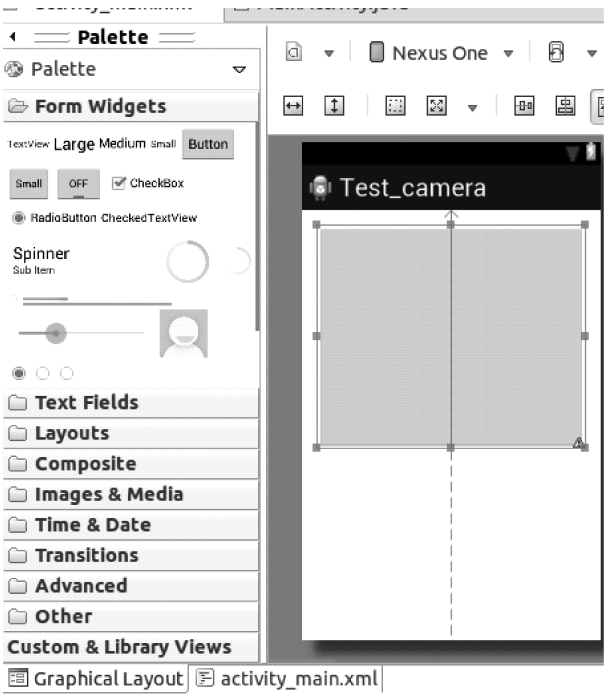


图 8-18　检查自定义组件

为了实现对远程监控的开启和停止操作，定义两个按钮分别实现功能。回到外层的 RelativeLayout 中，定义一个 < Button > 。

```
< Button
    android:id = "@ + id/start"
    android:layout_below = "@ id/inner_layout"
    android:layout_alignLeft = "@ id/inner_layout"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "开始监控"
    android:textSize = "12pt" / >
```

其中，below 属性将此 Button 组件设置为在嵌入的内层监控画面组件的下方，@ id 即为引用上述内层的 RelativeLayout 的 id；alignLeft 使其左侧与 inner_layout 对齐；text 为 Button 按

钮上显示的文字；textSize 为文字大小，pt 单位越大字符越大。

同理再定义一个 Button 来停止监控，id 命名为 stop，以便记忆。布局属性与上述 Button 代码类似，不同之处为使用 alignRight 来对齐 inner_layout 的右侧，以求界面美观；另将 text 换为“停止监控”，以示区别。

在图形视图中即可看见刚刚定义的两个 Button，如图 8-19所示，至此，本程序的布局定义完成。

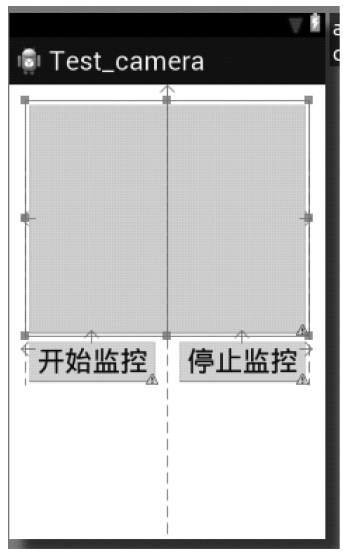


图 8-19　布局完成效果图

8.5.4　获取远程监控画面的代码实现

在工程目录下，找到 src 文件夹，此文件夹存放着应用程序的代码。打开定义的包名，找到 MainActivity 文件打开。

如图 8-20 所示，为了在代码中可以操作在布局文件中声明的组件，首先要建立三个变量，分别来操作 Layout 中的 ImageView 和两个 Button 组件。因为 MainActivity. java 默认是没有 import（引用）全部组件的 package（定义包）的，所以在定义 Button 型变量时，程序会报错，只需要将光标移动到报错位置，即 private Button 下，按照错误提示，选择 import android. widget. Button 即可，如图 8-20 所示，在 package 栏中新增了 import android. widget. Button。

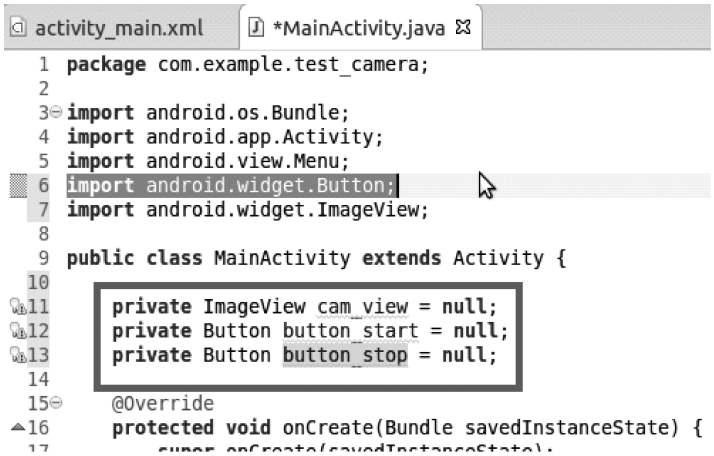


图 8-20　编辑 MainActivity. java

接下来使用 findViewById 方法，将定义的变量与 Layout 中的组件关联起来，如图 8-21 所示，通过引用组件的 id 名，分别实现了 ImageView 和 Button 各自的关联。注意使用 findViewById（id）方法，通过 id 名引用得到的对象，要使用强制类型转换，如（ImageView）和（Button），才可以相对应的不同类型的变量关联。

关于引用 R. id 组件名的用法，可打开 gen 目录下 R. java，如图 8-22 所示，使用@ + id 方法声明的组件，id 号即是在此处保存（此文件为系统自动生成，不可手动修改）。

```
activity_main.xml  *MainActivity.java
11      private ImageView cam_view = null;
12      private Button button_start = null;
13      private Button button_stop = null;
14
15      @Override
16      protected void onCreate(Bundle savedInstanceState) {
17          super.onCreate(savedInstanceState);
18          setContentView(R.layout.activity_main);
19
20          cam_view = (ImageView)findViewById(R.id.camera_view);
21          button_start = (Button)findViewById(R.id.start);
22          button_stop = (Button)findViewById(R.id.stop);
23
24      }
25
```

图 8-21　自定义变量与 Layout 组件的关联

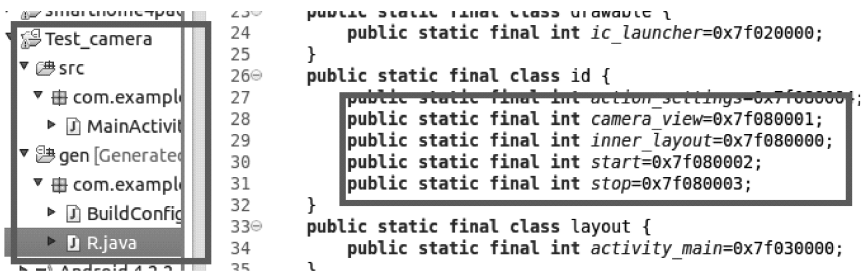


图 8-22　R. java 概览

在安卓系统中，程序的主线程中如果进行耗时长的操作，如网络操作、硬盘访问、数据库操作等，当超过一定时间未能响应用户操作，会导致应用程序的界面卡死，程序报错。因此，为了应用程序的线程安全性，把获取远程监控画面的操作放在一个单独的线程中。如图 8-23 所示在 MainActivity 中新建一个线程来完成获取远程监控画面的操作。

```
activity_main.xml  *MainActivity.java
11      private ImageView cam_view = null;
12      private Button button_start = null;
13      private Button button_stop = null;
14
15      @Override
16      protected void onCreate(Bundle savedInstanceState) {
17          super.onCreate(savedInstanceState);
18          setContentView(R.layout.activity_main);
19
20          cam_view = (ImageView)findViewById(R.id.camera_view);
21          button_start = (Button)findViewById(R.id.start);
22          button_stop = (Button)findViewById(R.id.stop);
23
24      }
25
26      Thread thread1 = new Thread(){
27
28          |
29      };
30
```

图 8-23　建立单独线程处理网络操作

在新线程的声明代码中单击鼠标右键，选择 source，选择 override/implement，选择线程的 run( ) 方法，重写来完成我们需要的操作，单击“OK”，如图 8-24 所示。

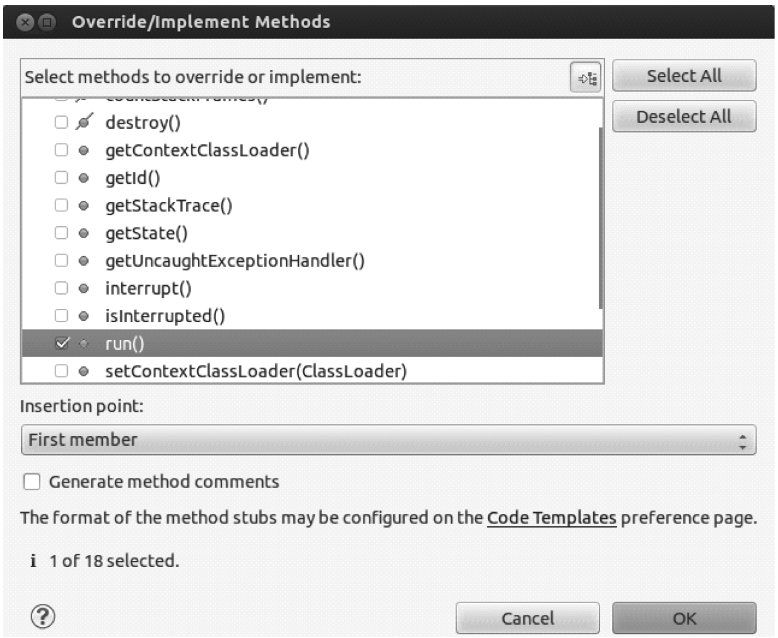


图 8-24　重写线程的 run( ) 方法

首先在 run( ) 方法中定义一个字符串变量 cam_url，这是已有的一个摄像头的网络访问地址。另外定义一个 Bitmap 变量，用来取得从摄像头获取的图像。

```
@ Override
        public void run( ) {
            // TODO Auto-generated method stub
            super. run( );

            String cam_url =
"http://115. 25. 48. 151:83/snapshot. cgi? user = admin&pwd = ";
            Bitmap bitmap = null;
```

然后在主类中定义一个 boolean 型作为控制网络操作的标识符，一个 URL 变量 getImage 来作为网络操作的对象，如图 8-25 所示。

```
17 public class MainActivity extends Activity {
18
19     private ImageView cam_view = null;
20     private Button button_start = null;
21     private Button button_stop = null;
22
23     private boolean flag = true;
24     private URL getImage = null;
```

图 8-25　定义网络操作的关键变量

在 thread1 的 run( ) 方法中，实现获取网络摄像头图像的操作，代码如下所示：使用上述定义的 flag 来控制网络访问，当允许访问时，执行 while( ) 循环。首先新建一个 URL，将 getImage 对象定义到 cam_url 保存的 URL 网址（这里使用 System. out. println( ) 方法打印信息，以便程序调试时观察代码执行进度)；随后创建了一个 HttpURLConnection 对象来连接到此 URL（关于安卓的 http 操作的 get 方法请自行学习)；定义一个输入流 is 来取得此http连接所得到的输入流，即远程摄像机的当前监控图像；最后对输入流按照 Bitmap 图像的格式进行解码，赋值给 bitmap，得到监控图像。

```
@ Override
public void run( ) {
    // TODO Auto-generated method stub
    super. run( ) ;

    String cam_url =
" http://115. 25. 48. 151:83/snapshot. cgi? user = admin&pwd = " ;
    Bitmap bitmap = null;

    while(flag)
    {
        try {
            getImage = new URL( cam_url) ;
            System. out. println( cam_url) ;
            } catch (MalformedURLException e) {
            e. printStackTrace( ) ;
            }
            try {
            HttpURLConnection conn =
(HttpURLConnection) getImage. openConnection( ) ;
            conn. setDoInput( true) ;
            conn. connect( ) ;
            InputStream is = conn. getInputStream( ) ;
            bitmap = BitmapFactory. decodeStream( is) ;
            is. close( ) ;
            } catch (IOException e) {
            e. printStackTrace( ) ;
            }
        }
    }
```

由于安卓系统只允许主线程更新 UI 界面，在 thread1 线程中获取到的图像，无法通过其本身的操作直接显示出来。所以要使用 handler 对象来实现主线程与非主线程的通信，将监

控画面传递到主线程，通过主线程显示在界面上。在 MainActivity 中定义一个子类 myhandler 继承 handler 类，它的构造函数的形参为 Looper 对象，用来实现线程间通信。

```
class myhandler extends Handler{
        public myhandler(Looper looper)
        {}
        @Override
        public void handleMessage(Message msg) {
            // TODO Auto-generated method stub
            Super. handleMessage(msg);
            cam_view. setImageBitmap((Bitmap)msg. obj);
        }
    }
```

在 myhandler 类中，重写 handleMessage 方法来实现对子线程 thread1 获取的监控画面的处理。将 bitmap 图像通过 msg. obj 对象传给 handleMessage 方法，然后此方法将其以 Bitmap 型变量重新取出，通过 cam_view 的 setImageBitmap 方法，显示在应用程序界面上。

在 thread1 的获取远程图像的 while( ) 循环中，使用 handler 的 obtainMessage （int，obj）方法，将获取的 bitmap 实时传递给主线程，如图 8-26 所示。

```
108                     bitmap = BitmapFactory.decodeStream(is);
109                     is.close();
110                     } catch (IOException e) {
111                     e.printStackTrace();
112                     }
113
114                     handler1.obtainMessage(0, bitmap).sendToTarget();
115             }
116
117         }
118
119     };
120
121
122     class myhandler extends Handler{
123
124         public myhandler(Looper looper)
125         {}
126
127         @Override
128         public void handleMessage(Message msg) {
129             // TODO Auto-generated method stub
130             super.handleMessage(msg);
131
132             cam_view.setImageBitmap((Bitmap)msg.obj);
133         }
134
135     }
```

两线程使用msg通信，传递bitmap图像

图 8-26　非主线程将实时图像发送到主线程

如图 8-27 所示，在 MainActivity 中对 Handler、Message、Thread 对象实例进行提前声明，避免程序中出现空指针（NullPointer）的错误：

```
31      private boolean flag = true;
32      private URL getImage = null;
33
34      private Handler handler1 = null;
35      private Message msg = null;
36      private Thread T1 = null;
37
38
39      @Override
40      protected void onCreate(Bundle sav
```

图 8-27 实例的提前声明

建立一个 OnClickListener 对象，来实现对不同 Button 按钮的功能进行定义。

```
OnClickListener Listener = new OnClickListener( ) {
    @ Override
    public void onClick( View v) {
        // TODO Auto-generated method stub
    }
}
```

在 Listener 对象的 onClick 方法中，添加 switch—case 分支语句，根据不同 Button 的 id 号不同，来判断当前单击的 Button，触发各自对应的功能：

```
@ Override
        public void onClick( View v) {
            // TODO Auto-generated method stub
        switch( v. getId( ) ) {
        case R. id. start:
            if( T1 ! = null)
            {
                if( T1. isAlive( ) )
                {return;}
            }else
            {
                if( handler1 = = null)
                {
                HandlerThread handlerThread = new
HandlerThread( "handler_thread" ) ;
                handlerThread. start( ) ;
                handler1 = new myhandler( handlerThread. getLooper( ) ) ;
                }
            }
            flag = true;
```

```
            T1 = new Thread( thread1 ) ;
            T1. start( ) ;
            break;
        case R. id. stop:
            flag = false;
            break;
        default:
            break;
        }
```

单击开始监控的 Button 后，首先判断 Thread1 的实例 T1，如果在运行，则不做处理。否则判断主线程负责通信的 myhandler 的实例 handler1，如果为空则对其实例化，以便获取子线程 T1 传送的图像；最后将标识符 flag 置 true，准备开始获取远程图像的 while（flag）循环，将 T1 实例化，开始执行。

如果单击停止监控的 Button，flag 置 false，由此 T1 不再执行 while（flag），监控停止，run( ) 方法执行完毕，T1 被安卓系统终结。直到再次单击开始监控的 Button，T1 重新被实例化。

随后将此 Listener 对象绑定到 Button 对象上，如图 8-28 所示。

```
46          button_stop = (Button)findViewById(R.id.stop);
47
48          button_start.setOnClickListener(Listener);
49          button_stop.setOnClickListener(Listener);
50
51      }
52
53      OnClickListener Listener = new OnClickListener() {
54
```

图 8-28　绑定 Listener 对象到 Button 组件

最后，打开工程目录下的安卓 Manifest. xml 文件，此文件定义了应用程序的配置信息、权限等。如图 8-29 所示，为程序添加网络访问的权限。

```
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.test_camera"
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7    <uses-permission android:name="android.permission.INTERNET" />
8
9      <uses-sdk
10         android:minSdkVersion="9"
11         android:targetSdkVersion="16" />
```

图 8-29　添加网络访问权限

8.5.5 调试与实现

至此，应用程序的功能已经开发完成。将安卓手机用 USB 线缆连接至 PC，打开确保已在手机设置中打开 USB 调试模式；在 PC 端新打开一个终端窗口，输入："adb devices"。如果不能识别出已连接的安卓设备，终端输入："sudo adb kill-server"，关闭 adb 服务时要输入管理员密码，然后输入"sudo adb start-server"命令启动 adb 服务。启动后，就需要用"adb devices"命令查询当前有多少模拟器/设备在线情况。此时即可识别出当前连接的安卓设备；否则重复上述过程或重新启动 Eclipse 开发环境。

识别出设备后，右键单击工程总目录，选择：Run as→Android Application，按提示选择已连接的设备，单击"确定"，程序即可在线下载到安卓手机并运行。

确认手机可连接到远程摄像头所在的网络后，开启监控，效果如图 8-30 所示：

图 8-30 程序效果图

8.6 思考题

1）调试过程中遇到哪些问题？如何解决？
2）在没有安卓终端的情况下是否可以调试安卓应用程序，借助什么工具？
3）尝试在 Windows 环境下完成安卓开发环境配置。
4）查找资料，对比 IOS 与安卓开发的不同。

8.7 实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤以及调试过程、心得体会，最后要附上程序，并按要求回答思考题 1）、2）、3）、4）。

参 考 文 献

[1] 公磊，周聪. 基于 Android 的移动终端应用程序开发与研究［J］. 计算机与现代化，2008（08）.
[2] 林永军. Android 课程实验教学系统的研究与设计［D］. 北京：北京邮电大学，2012.
[3] 刘振宇，周荣慧. Google 服务在 Android 上的运用与分析［J］. 软件导刊，2010（11）.
[4] 靳建林，刘晓燕. 基于 Android 系统的无线监控系统［J］. 软件导刊，2011（06）.
[5] 宋小倩，周东升. 基于 Android 平台的应用开发研究［J］. 软件导刊，2011（02）.
[6] 耿东久，索岳，陈渝，等. 基于 Android 手机的远程访问和控制系统［J］. 计算机应用，2011（02）.

[7] 汪宇，吕卫，杨博菲，等. 基于 Android 平台的智能家居监控系统［J］. 电视技术，2012（02）.

[8] Paul Hyde. Java 线程编程［M］. 周良忠，译. 北京：人民邮电出版社，2003.

[9] 顾聪，陈益强，刘军发，等. 基于 Android 平台的室内 LBS 系统设计与实现［J］. 计算机工程与设计. 2012（01）.

[10] Michael Juntao Yuan. J2ME 移动应用程序开发［M］. 梁超，王延华，译. 北京：清华大学出版社，2004.

第 9 章　移动终端之 Win8 平板电脑开发实验

9.1　实验目的

1）了解使用 Win8 平板电脑实现智能家居电灯控制的方法和原理。
2）掌握 Win8 平板电脑应用程序的基本开发流程。
3）熟悉 Win8 平板电脑应用程序开发环境 Win8\VS2012。

9.2　实验内容

在 Win8 环境下，搭建 Win8 平板电脑下的智能家居控制系统，实现智能家居电灯控制功能，达到掌握原理、熟悉应用的效果。

9.3　实验所用仪表及设备

1）硬件：PC、网络摄像头、Win8 平板电脑。
2）软件：Win8 操作系统、Visual Studio Express 2012 for Windows 8。

9.4　实验原理

9.4.1　系统简介

Win8 是由微软公司开发的、具有革命性变化的操作系统。该系统旨在让人们的日常电脑操作更加简单和快捷，为人们提供高效易行的工作环境。Win8 支持来自 Intel、AMD 和 ARM 的芯片架构。Windows Phone 8 采用和 Win8 相同的 NT 内核并且内置诺基亚地图。2011 年 9 月 14 日，Win8 开发者预览版发布，宣布兼容移动终端，微软将苹果的 IOS、Google 的安卓视为 Win8 在移动领域的主要竞争对手。北京时间 2012 年 8 月 2 日凌晨消息，微软宣布 Win8 开发完成，正式发布 RTM 版本；2012 年 10 月 26 日正式推出 Win8，微软触摸革命开始。Win8 桌面如图 9-1 所示。

9.4.2　系统架构

Win8 支持个人电脑（X86 构架）及平板电脑（X86 构架或 ARM 构架）。Win8 大幅改变以往的操作逻辑，提供更佳的屏幕触控支持。新系统画面与操作方式变化极大，采用全新的 Metro（新 Windows UI）风格用户界面，各种应用程序、快捷方式等能以动态方块的样式呈现在屏幕上，用户可自行将常用的浏览器、社交网络、游戏、操作界面融入。

图 9-1　Win8 桌面

9.4.3　系统优化

除性能提高、用户体验更舒适外，Win8 更方便开发者使用。

Windows 应用商店帮助开发人员将自己的应用程序销售到全球各地，只要有 Win8 的地方，都可以向用户展示你开发的应用。

Win8 允许开发人员使用现有的语言进行编程，支持 C、C ++、C#、VB、HTML 和 CSS、JavaScript、XAML 等。

Win8 支持 DirectX 11，开发人员可以轻松创建全屏游戏，并且获得流畅的操作体验。

Win8 中的诸多创新功能都和分布式文件系统复制（DFSR）服务有关，多个复制引擎将会通过多个服务器执行工作进而简化文件夹同步过程。从这里可以看出，这一功能改进不仅涉及 Win8 客户端还和 Win8 服务器有关。

9.4.4　开发工具

Win8 平板电脑的开发前提需要搭建 Win8 系统环境，并安装 Microsoft Visual Studio Express 2012 for Windows 8。VS2012 是构建 Windows 应用的工具，它包括 Win8 SDK、Blend for Visual Studio 和项目模板，但是首先必须安装 Win8。

您是否拥有平板电脑无关紧要，Visual Studio 包括一个在平板电脑上体验您的应用的工具。选择部署到您本地的机器进行无系统窗口控件的应用程序（chrome-less），或者部署到模拟器以测试便携功能，或者在平板电脑上远程调试应用程序，以获得实际体验。

在开始之前，还需要具备开发者许可证，您可以通过 Visual Studio 首次运行时获取。

9.5 实验程序设计

9.5.1 开发应用程序的步骤

1）创建用户界面：了解如何为应用创建用户界面。

2）定义应用资源：了解如何定义应用的资源以改善可维护性和本地化。

3）响应用户交互：了解如何处理来自触控输入、鼠标和键盘设备的用户交互。

4）使用数据和文件：了解如何绑定数据、读取、写入和保存到文件，以及如何操纵 XML 数据。

5）连接到对等端、Web 和网络服务：了解如何创建连接的应用。连接的应用或网络感知的应用可以将网络用作多种用途，包括 RSS 源、游戏以及用于与附近设备交互。

6）管理用户信息：了解如何使用漫游凭据，如何使用 Live 服务验证用户，以及如何设置单一登录。

7）启动和恢复应用：了解如何启动、挂起和恢复应用，同时保持应用数据的安全和最新。

8）添加多媒体：了解如何捕获多媒体，播放音频和视频，处理图像文件，共享或流式传输媒体以及代码转换多媒体。

9）集成设备、打印机和传感器：了解如何支持打印机、相机、传感器、可移动存储等设备。还可以了解如何为游戏选择正确的运动和方向传感器，如何使用光传感器调整屏幕亮度，以及如何检测用户的地理位置。

10）创建 Windows 运行时组件：了解如何使用 C ++ 、C# 或 Visual Basic 创建组件（基本是 DLL），然后从使用 JavaScript 的 Windows 应用商店应用中以简单自然的方式调用这些组件。

11）全球化你的应用：了解如何本地化/全球化你的应用。

12）使你的应用可访问：了解如何创建受众最广泛（包括残障人士）的应用。

13）将应用打包：了解如何使用 Visual Studio 访问 Windows 应用商店并将你的应用打包以供分发。

14）调试和测试应用：了解如何使用 Visual Studio 中的测试和调试工具。

功能流程及基础交互设计如图 9-2 所示。

9.5.2 新建项目

在 Win8 系统下，打开已经安装成功并具备开发者权限的 Microsoft Visual Studio Express 2012 for Windows 8，新建一个项目，如图 9-3 所示。

模板选中 C#下“空白应用程序（XAML）”，名称为“Light_Ctrl”，如图 9-4 所示。

9.5.3 添加服务引用

项目创建成功后，可以在“解决方案资源管理器”中找到“引用”→“添加服务引

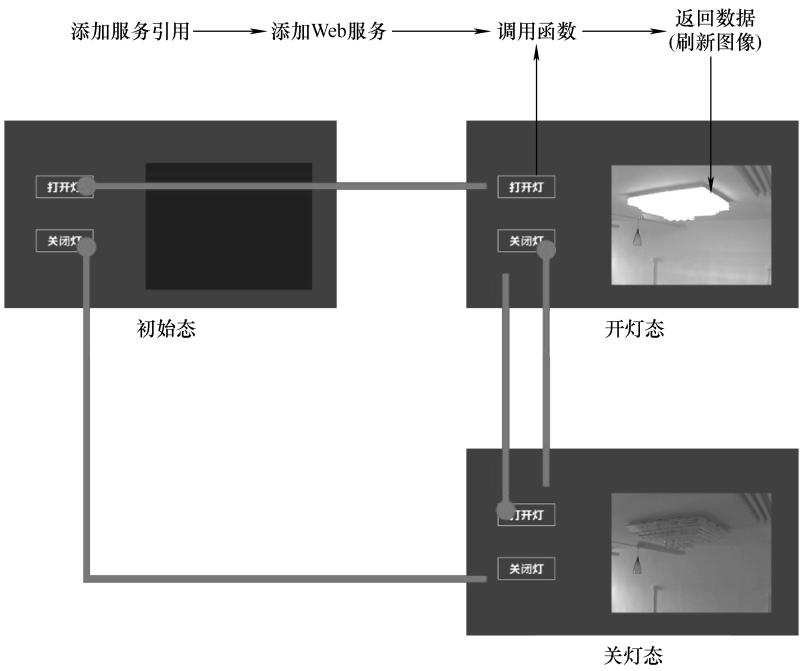


图 9-2　功能流程及基础交互设计

图 9-3　新建项目

用”，如图 9-5 所示。

输入地址“http://115.25.48.254:8902/BasicGateway/”，单击“转到”。(此地址为

图 9-4 选中模板并命名

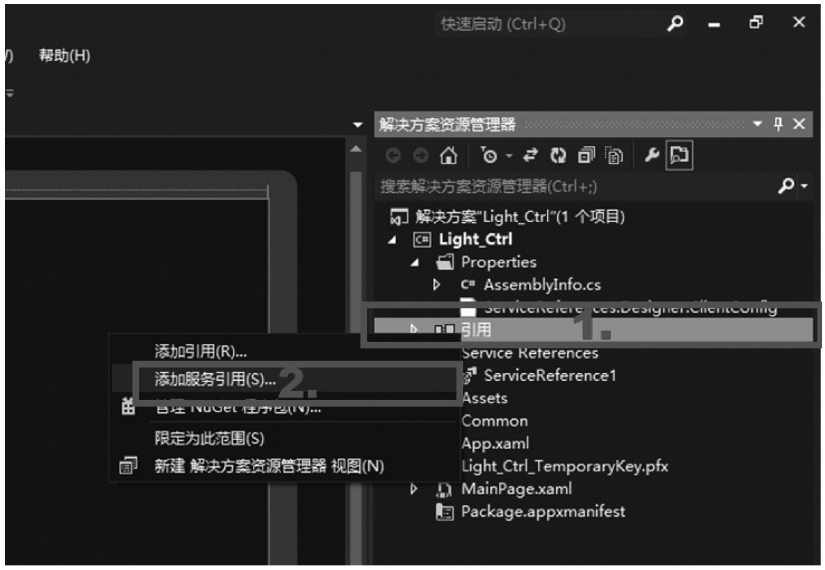


图 9-5 添加服务引用（1）

Web 服务地址），单击“确定”退出，如图 9-6 所示。

9.5.4 创建用户界面

在解决方案资源管理器中找到“MainPage. xaml”，双击打开。在“工具箱”中找到

图 9-6　添加服务引用（2）

"Button"，拖入用户主界面上，如图 9-7 所示。

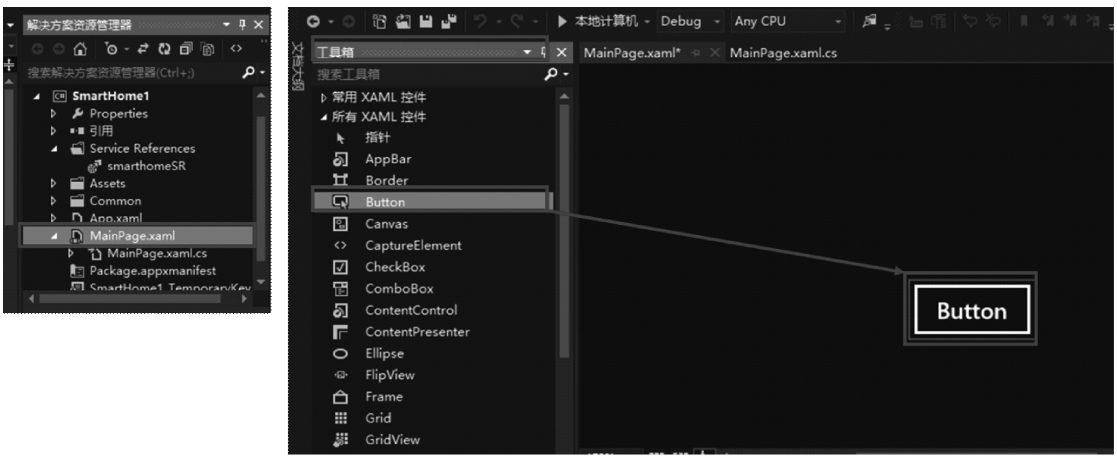


图 9-7　添加控件- Button

按照上述方法在主界面中添加两个"Button"控件和一个"Image"。排列方式如图 9-8 所示。

对控件属性及触发处理进行设置，具体见表 9-1 和图 9-9 所示。

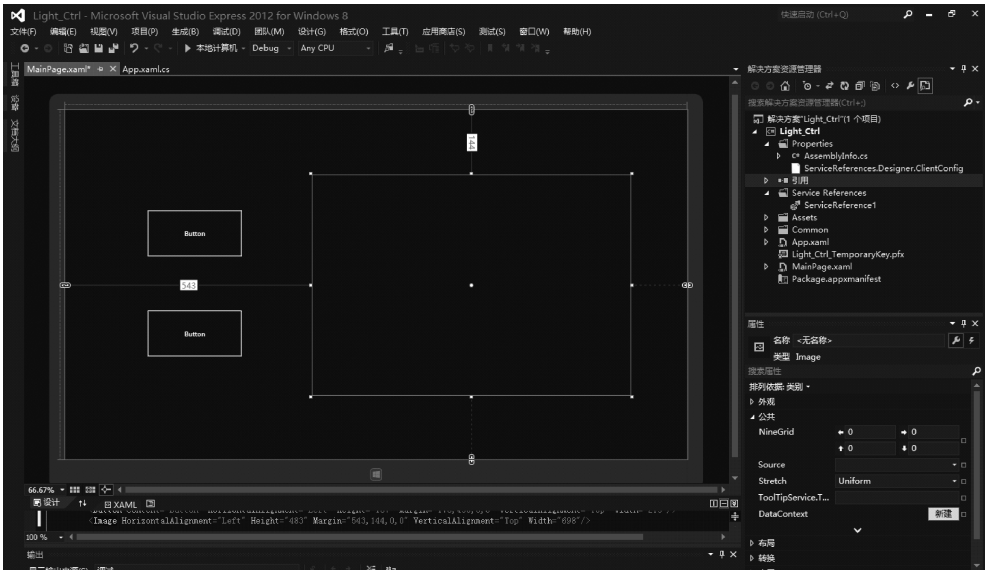


图 9-8 用户界面控件布局

表 9-1 控件属性

控件	名称	属性		触发处理
		公共 . Content	文本 . 字号	Click
Button1	OnBtn	打开灯	50px	onBtn_Click
Button2	OffBtn	关闭灯	50px	offBtn_Click
Image	image			

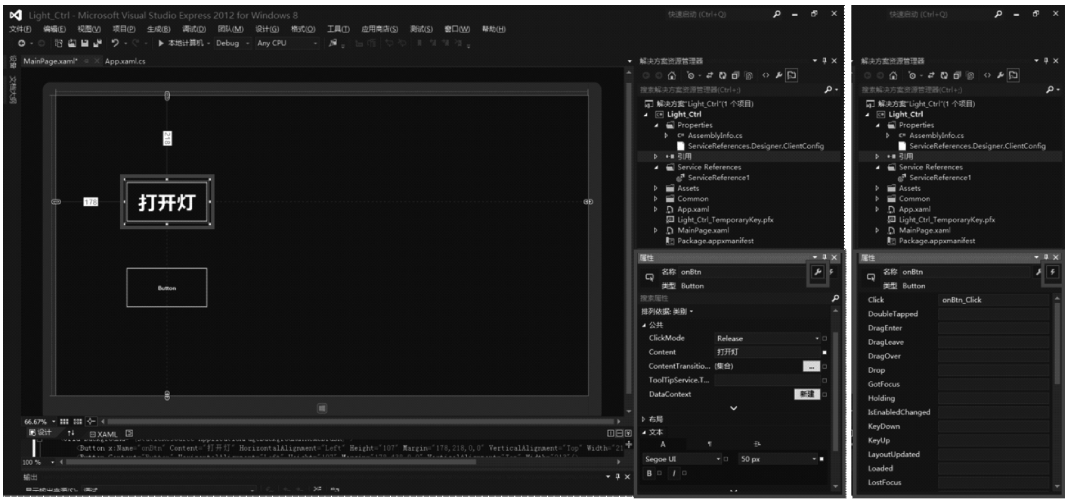


图 9-9 控件属性设置_Button

9.5.5 代码实现

双击控件，进入后台“MainPage. xaml. cs”。添加引用，具体如下：

```
using Light_Ctrl. ServiceReference1;  //添加服务引用
using System. Net;
using Windows. Storage. Streams;
using Windows. UI. Core;
using Windows. UI. Xaml. Media. Imaging;
```

在“public sealed partial class MainPage : Page”与“public MainPage( )”之间定义一个 BasicGatewayServiceClient 对象，来完成各种操作。

BasicGatewayServiceClient 通过添加 Web 服务获得，WebService 地址为 http://115.25.48.254:8902/BasicGateway/

```
BasicGatewayServiceClient client = new BasicGatewayServiceClient( );
```

在“public MainPage( )”中定义 image 的初始状态。程序一开始便获取主灯视频图像。(考虑到 asf 视频格式目前无法在 Win8 上解码，并且如果用连续图像模拟视频的话需要图像格式的转换且增加流量，故本实验刷新获取一张静态图片来展示开关灯的操作结果。)

```
BitmapImage bitmap = new BitmapImage( );
bitmap. UriSource = new
Uri(String. Format("http://115. 25. 48. 254:8902/RestfulService/Devices/
WebCamera3/Photo? clientId = phone&next_url = " + Guid. NewGuid( )));
image. Source = bitmap;
```

通过调度智能家居服务器端电灯控制函数 client. InvokeAgentOperationAsync( )，实现 OnBtn_Click点击事件触发效果。实现电灯的物理端实际开灯操作后，刷新摄像头抓取的主灯画面，使得操作结果立即可见。具体实现如下：

```
private void onBtn_Click(object sender,
Windows. UI. Xaml. RoutedEventArgs e)
{
//开灯操作
try
    {
    //Win8 为设备 Id,MainLamp 为设备名称,Open 为设备操作名,parass 为操作参数(目前为空,可以如下定义:
    //Dictionary < string,AgentOperParam >  parass = new
Dictionary < string,AgentOperParam > ( );)
    Dictionary < string,AgentOperParam >  parass = new Dictionary < string,
AgentOperParam > ( );
    client. InvokeAgentOperationAsync("Win8","MainLamp","Open",parass);

    //更新图像
    BitmapImage bitmap = new BitmapImage( );
    bitmap. UriSource = new
```

```
Uri(String.Format("http://115.25.48.254:8902/RestfulService/Devices/
WebCamera3/Photo?clientId=phone&next_url="+Guid.NewGuid()));
        image.Source=bitmap;
        }
catch (Exception)
        {
        }
}
```

关灯操作与开灯操作类似，调度 client.InvokeAgentOperationAsync() 函数时更新参数设置，第三个参数由“Open”更新为“Close”即可。具体实现如下。

```
private void offBtn_Click(object sender,
Windows.UI.Xaml.RoutedEventArgs e)
{
//关灯操作
try
        {
        //相应地将操作名改为 Close 即可
        Dictionary<string,AgentOperParam> parass=new Dictionary<string,
AgentOperParam>();
        client.InvokeAgentOperationAsync("Win8","MainLamp","Close",
parass);

        //更新图像
        BitmapImage bitmap=new BitmapImage();
        bitmap.UriSource=new
Uri(String.Format("http://115.25.48.254:8902/RestfulService/Devices/
WebCamera3/Photo?clientId=phone&next_url="+Guid.NewGuid()));
        image.Source=bitmap;
        }
catch (Exception)
        {
        }
}
```

9.5.6 调试实现

VS2012 中 Win8 平板电脑开发可以通过三种方式调试实现：模拟器、本地计算机、远程计算机。本实验使用模拟器调试，错误列表如图 9-10 所示，测试运行结果如图 9-11 所示。

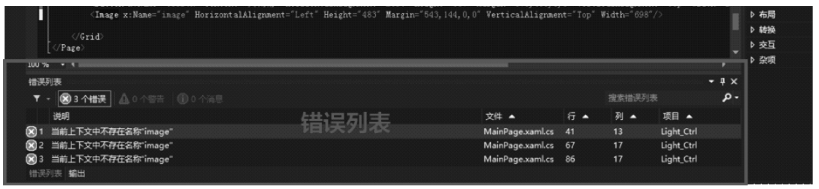


图 9-10　错误列表

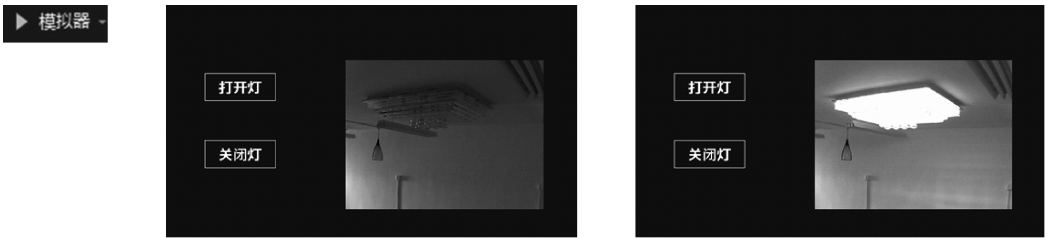


图 9-11　测试运行结果

9.6　思考题

1）调试过程中遇到哪些问题？如何解决？

2）简述开发流程。

3）如何改善 UI 设计，增强界面友好性？

9.7　实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤以及调试过程、心得体会，最后要附上程序，并按要求回答思考题 1）、2）、3）。

参 考 文 献

[1] 伏英娜. Windows Phone 7 应用开发指南 [M]. 北京：电子工业出版社，2011.

[2] PETZOLD C. Programming Windows Phone 7 [M]. Redmond：Microsoft Press，2010.

[3] Yochay. Understanding Microsoft Push Notifications for Windows Phone. WindwosTeamblog. 2010.

[4] 朱宏等. 超越想象——Windows 8 应用设计与开发 [M]. 北京：电子工业出版社，2013.

[5] 周家安. Windows phone8 开发技巧与案例精解 [M]. 北京：机械工业出版社，2013.

[6] 林政. 深入浅出：Windows Phone 8 应用开发 [M]. 北京：清华大学出版社，2013.

第 10 章　健康物联网

10.1　实验目的

1）了解 2.4GHz 无线通信技术及其数据传输协议的特点。

2）了解 Socket，掌握使用 Socket 来实现信息发送和接收的过程和实现方式，能够根据提供的接口编写出对应需求的代码。

10.2　实验内容

在 Win7 平台下，开发基于 Visual Studio 2010 开发环境及 SQL Server 2008 数据库的 PC 端健康监护软件，和基于安卓操作系统的手机客户端软件，实现将健康设备收集到的监测数据存入数据库，并且对历史数据库进行查看、数据图表分析及获取健康评估和建议的功能，使用户能有效管理和维护自身的健康状况。

10.3　实验所用硬件及软件

10.3.1　硬件设备

1）健康监测设备一套：用于监测用户生理数据如图 10-1 所示。

2）服务器 1 台：用于接收、解析、存储用户基本信息和监测数据。对开发环境的计算机环境而言至少需要满足以下要求：

① 2.0（GHz）及以上处理器。

② 内存要求：2GB 或以上。

③ 显卡要求：512MB 或以上显存的显卡。

④ 硬盘要求：推荐 250GB 或以上。

3）安卓智能手机：监测及上传监测数据。

10.3.2　系统所用软件

1. 系统开发环境

PC 端：

1）操作系统要求：Win7 等。

2）开发环境：Eclipse + Android SDK + ADT，Visual Studio 2010，SQL Server 2008。

手机端：

操作系统要求：安卓 4.0 及以上。

设备名称	设备图片	功　　能
人体秤		• 测量体重 • 2. 4GHz 无线传输（健康数据管理）
脂肪仪		• 测量人体脂肪率、水分、肌肉量、内脏脂肪 • 2. 4GHz 无线传输（健康数据管理）
血压计		• 测量血压、脉搏、心律不齐提示 • 2. 4GHz 无线传输（健康数据管理）
血氧仪		• 测量血氧饱和度 • 2. 4GHz 无线传输（健康数据管理）
血糖仪		• 测量血糖 • 2. 4GHz 无线传输（健康数据管理）
耳温枪		• 测得人体体温 • 2. 4GHz 无线传输（健康数据管理）
2. 4GUSB 接收器		• 测量产品与计算机或是显示器之间的数据接收功能

图 10-1　健康监测设备

2. 软件成品

PC 端：

1）数据接收、解析及入库软件。

2）监测数据图表分析、健康评估软件。

手机端：“智能家居” App。

10.4 实验原理

健康监测设备将测得的用户生理数据通过 2.4GHz 无线通信方式按约定的数据格式发送出去，数据通过 2.4GUSB 接收器被服务器接收。服务器将接收到的数据发送给向其请求监测数据的手机端软件，用户可以通过手机端软件查看测量状况，并授权数据上传到数据库。服务器接收到手机端发回的反馈信号后，会对数据进行解析和存储到数据库的操作。

用户可以运行监测数据图表分析及健康评估软件对历史数据进行查看，并获得健康测评意见。

10.5 相关技术

1. Socket 通信

Socket 通常也称为“套接字”，应用程序通常通过“套接字”向网络发出请求或者应答网络请求。使用 Socket 完成数据通信的流程图如图 10-2 所示：

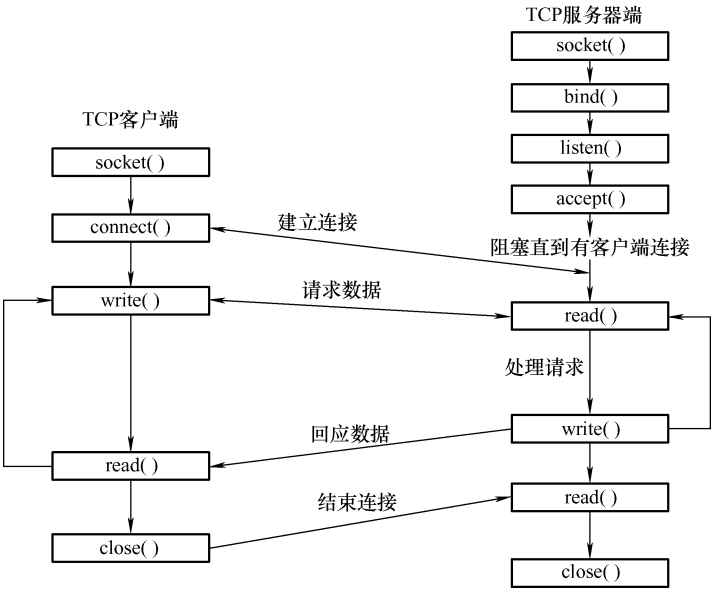


图 10-2 Socket 通信

用到的类包括 IPAddress、Dns、Socket、IPEndPoint 等，利用 TCP 编程如下：

（1）服务器端

1）建立 TCP 监听器 TcpListener 对象。

```
IPEndPoint ipep = new IPEndPoint(IPAddress.Any,9528);
TcpListener lisner = new TcpListener(ipep);
```

2）启动监听器。

```
lisner. Start( );
```

3）用监听器获取连接进来的套接字（Socket）。

```
Socket client = lisner. AcceptSocket( );
```

4）通过 Socket 的 Receive 方法获取客户端发送的数据。

```
byte[ ]a = ReceiveData( client,11 );
    ……
public static byte[ ]ReceiveData( Socket s,int size)
{
    int total =0;
    int dataleft = size;
    byte[ ]data = new byte[ size ];
    int recv;
    try
    {
        while ( total < size)
        {
            recv = s. Receive( data,total,dataleft,SocketFlags. None);
            if ( recv = =0)
            {
                data = null;
                break;
            }
            total + = recv;
            dataleft - = recv;
        }
    }
    catch (System. Exception) { }

    return data;
}
```

5）通过 Socket 的 Send 方法向客户端发送数据。

```
IPEndPoint ipep = new IPEndPoint( IPAddress. Any,9527);
TcpListener lisner = new TcpListener( ipep);
lisner. Start( );
……
Socket client = lisner. AcceptSocket( );
……
client. Send( temp,temp. Length,0);  //temp 为需要传输的字节数组数据
```

6）在通信结束后，释放资源结束监听。

（2）客户端

1）建立 TCP 客户端 TcpClient 对象。

2）连接服务器。

3）获得客户端网络传输流。

4）通过 Stream 的 Write 方法向服务器端发送的数据。

5）通过 Stream 的 Read 方法读取服务器端发来的数据。

6）在通信结束后，需要释放资源，结束和服务器的连接。

2. DevEXPRESS 控件

健康历史监测数据的图表分析功能的开发部分用到了 DevEXPRESS。利用 DevEXPRESS 控件，如图 10-3 所示，可以方便地读取数据库数据，绘制 2D/3D 曲线、柱状、饼状图，且

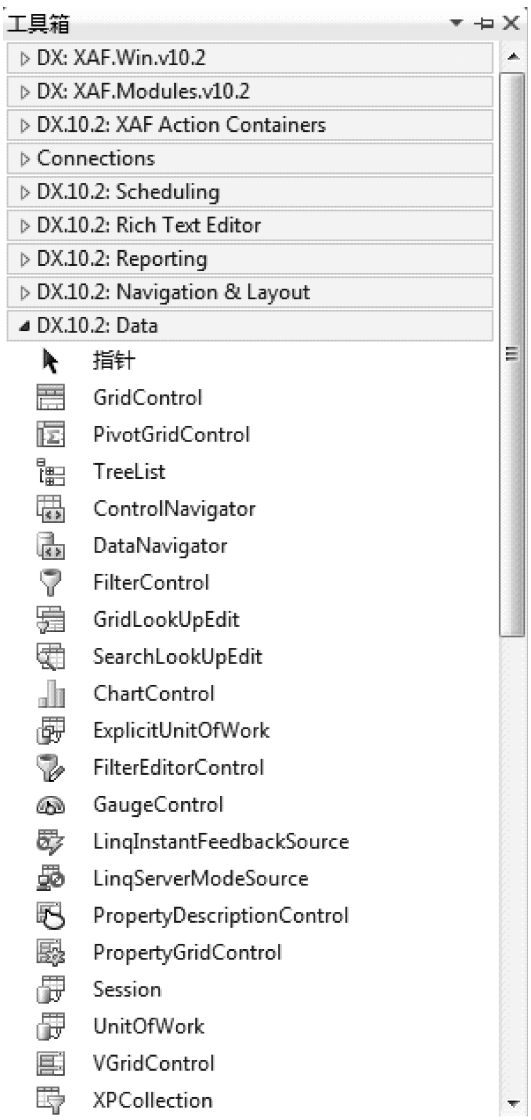


图 10-3　DevEXPRESS 控件

有丰富的表现形式。

10.6　实验步骤

10.6.1　搭建运行环境

1. 安装 SQL Server 2008

2. 安装 Visual Studio 2010

3. 连接数据库服务器

打开 SQL Server Management Studio，弹出如图 10-4 所示对话框。

图 10-4　连接数据库服务器

1）服务器名称一栏填写连接的数据库所在的 IP 和数据库实例名，格式：IP\实例名。若连接到本地的数据库实例，可以写成上述样式，其中“.”代表本机，SQLEXPRESS 是本机上的一个实例名。

2）身份认证选项有“Windows 身份认证”和“SQL Server 身份认证”两种认证模式。连接本机数据库建议选择“Windows 身份认证”，如图 10-5 所示。选择“SQL Server 身份认证”需填写登录名和密码，如图 10-6 所示，多用于配合服务器名称一栏的内容连接到远程数据库服务器。

4. 添加数据库

在“数据库”上单击鼠标右键，选择“附加”，如图 10-7 所示。

在弹出来的窗体中选择“添加”，如图 10-8 所示。

找到需要添加的数据库文件，单击“确定”，如图 10-9 所示。

图 10-5 选择 Windows 身份验证

图 10-6 选择 SQL Server 身份验证

数据库被添加进来，如图 10-10 所示。

10.6.2 修改代码

1）运行 Visual Studio，打开 MiBody361 工程文件，如图 10-11 所示。

图 10-7　选择附加

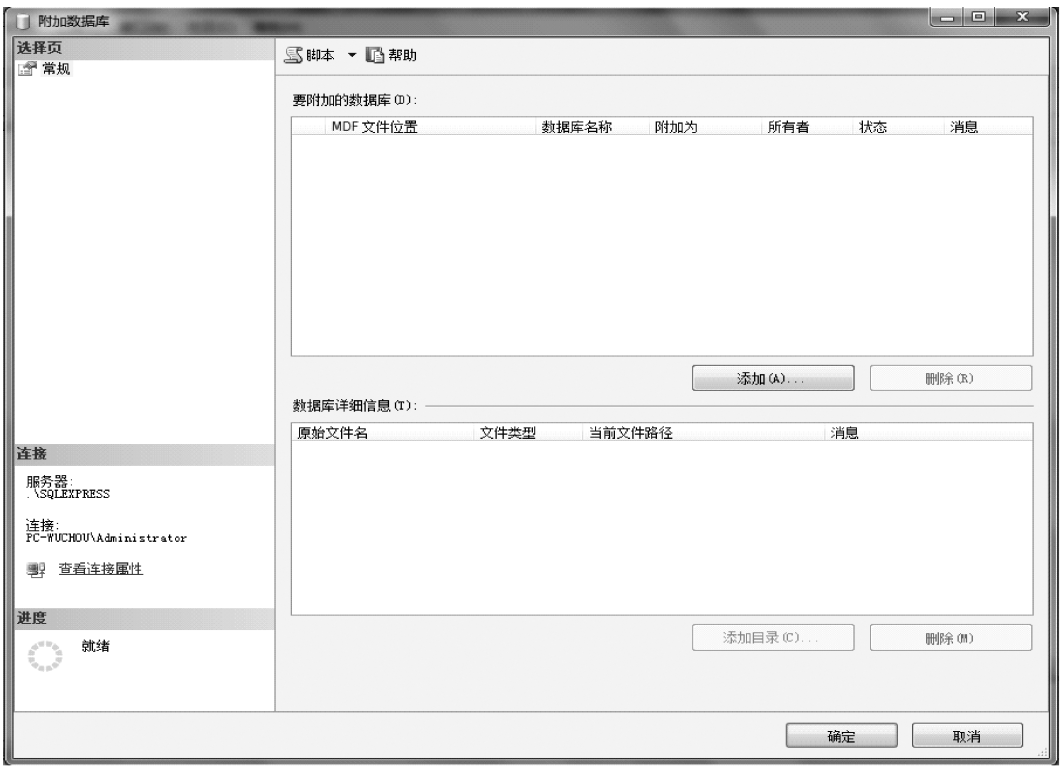


图 10-8　选择添加

打开 Homepage. cs 代码，如图 10-12 所示。

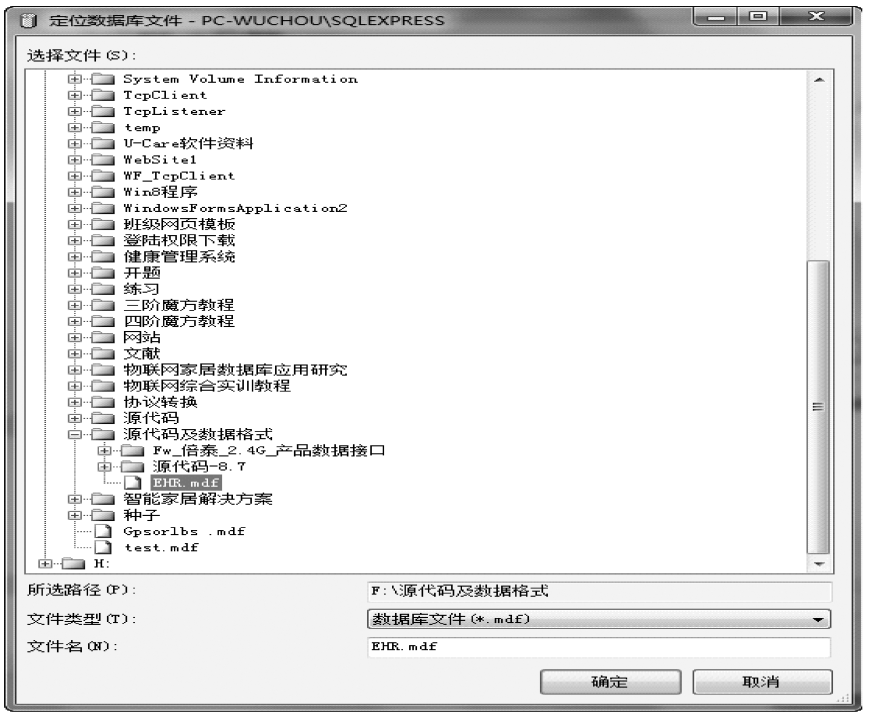


图 10-9 找到添加的数据库

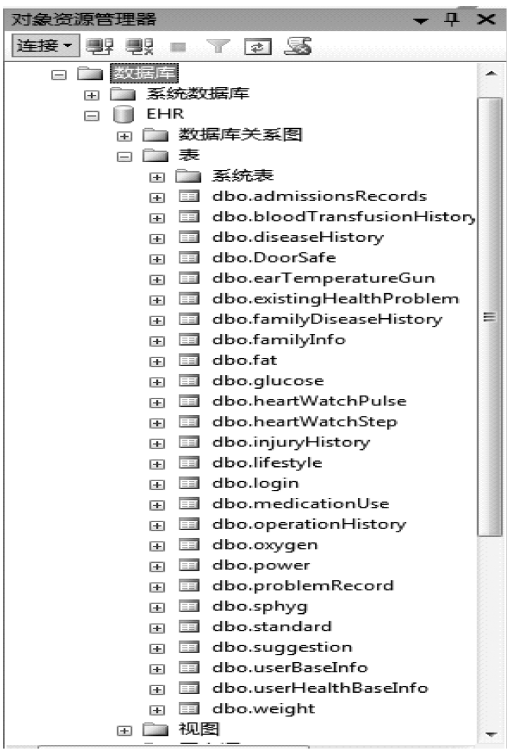


图 10-10 已添加的数据库

图 10-11　打开 MiBody361 文件

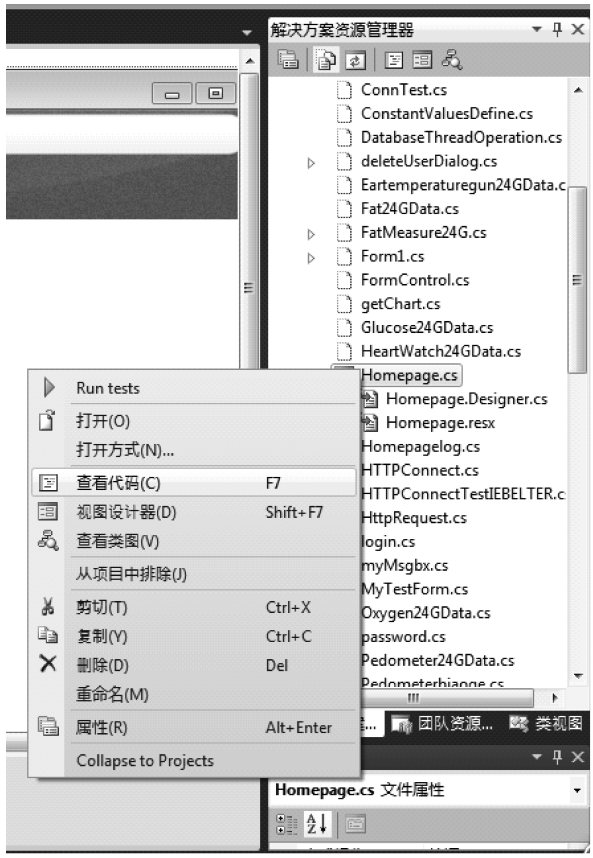


图 10-12　查看 Homepage. cs 代码

修改程序与数据库连接的代码，使健康监测数据能顺利写入数据库。

```
//    string strcon = @"Data Source=" + "192.168.1.29" + "" + "" + ";" + "Network Library=
string strcon = @"Server=.\SQLEXPRESS;Database=EHR;Integrated Security=True";
```

编译改过的程序。

2）运行 Visual Studio，打开 bloodpressuretest 工程文件，如图 10-13 所示。

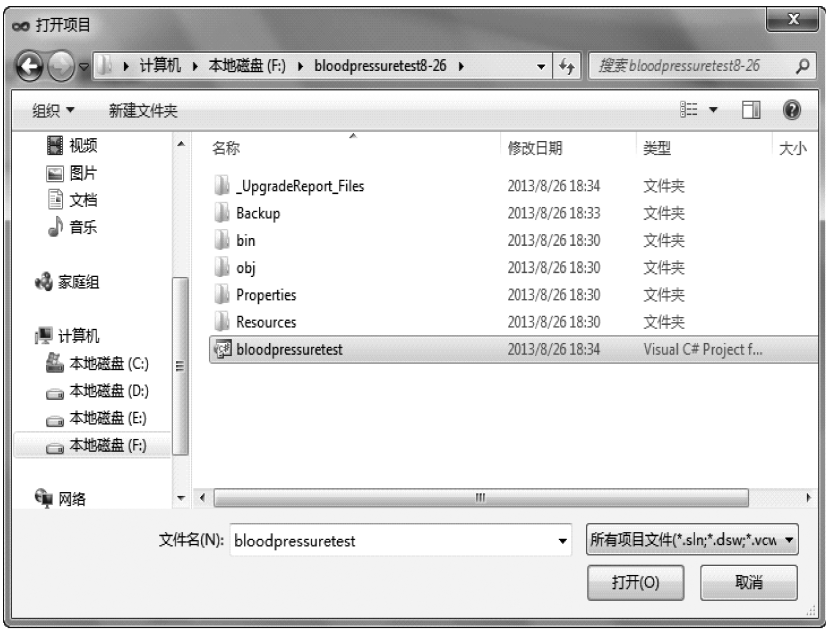


图 10-13　打开 bloodpressuretest 工程文件

查看 Form1. cs 代码，如图 10-14 所示。

修改程序与数据库连接的代码，使程序能顺利调用数据库数据。

```
mysqlconnection = new SqlConnection(@"Server=.\SQLEXPRESS;Database=EHR;Integrated Security=True");
```

编译改过的程序。

10.6.3　系统使用说明

1）运行服务器端的数据接收、解析及入库软件和数据库。

2）将 2.4GUSB 接收器插到 vPC 服务器 USB 口上。

3）在手机端安装“智能家居” App。设置手机 WLAN 连接到与 vPC 连接到相同的局域网中。运行软件。单击“控制面板”进入控制页面，如图 10-15 所示，单击“系统设置”，在“健康终端 IP 地址”一栏输入 vPC 的 IP 地址（可在 vPC 的命令提示符里输入命令 ipconfig 查看 vPC 的 IP 地址），配置“健康终端端口号”为 9527，单击“确定”以保存设置，如图 10-16 所示。

4）单击手机“智能家居”软件控制页面上的“健康检测”，进入到健康数据接收页面。启动倍泰健康监测设备，单击健康数据接收页面中的“获取数据”并在“用户列表”中选择监测用户，当监测数据项信息稳定且完整显示在页面上时，如图 10-17 所示，单击“断开连接”，“上传结果”，将数据提交到数据库。

图 10-14　查看 Form1. cs 代码

图 10-15　初始界面及控制页面

5）运行监测数据图表分析、健康评估软件如图 10-18 和图 10-19 所示。单击界面上“链接数据库”按钮，与数据库建立连接。在“查询 ID”、“查询时间”里输入要查看的监测对象的 ID 及要查看的历史数据范围，单击“查询”。

注意事项：

1）数据采集、入库及历史数据查看必须配置智能手机及 PC，服务器在同一个局域网

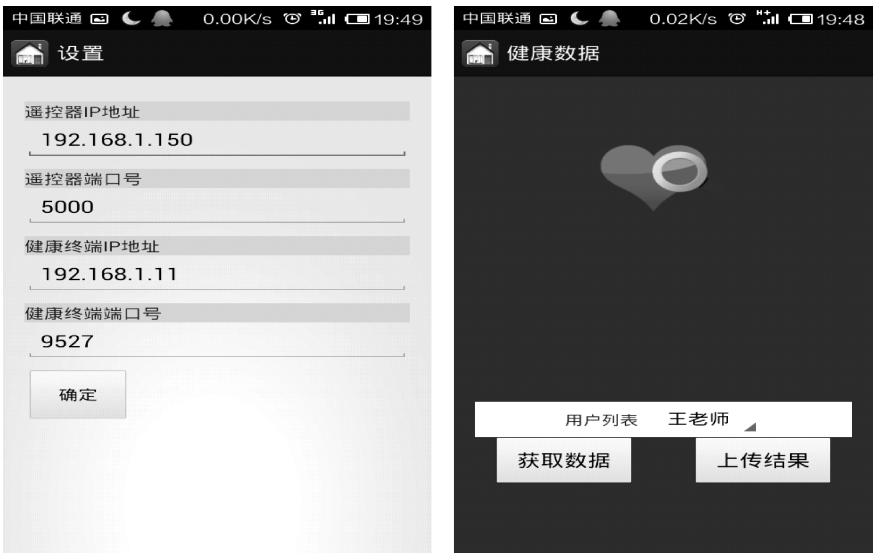


图 10-16 设置页面及健康数据采集页面

图 10-17 获取血氧仪数据及脂肪仪数据

中，并且配置好相应的项。

2）人体秤数据、耳温枪在体重稳定时传输一次，所以最好在测量之前单击“获取数据”。

脂肪仪需采集 4 个数据项，并且 4 个数据项也只传输一次，也需要在测量之前单击“获取数据”。若数据项信息不完整，可先返回至“控制页面”，再重新测量。

血氧仪数据会每秒更新一次数据，等到测量数据趋于稳定后再单击“获取数据”。

血压计需在人体安静状态下以坐姿测量右手胳膊，并使测量位置与心脏保持在同一水平

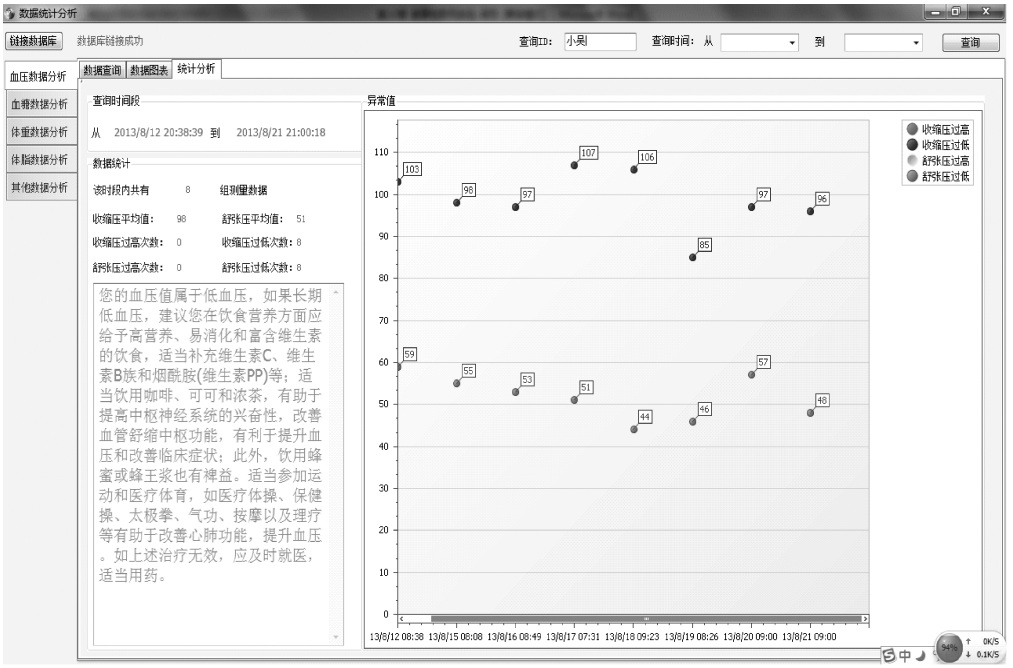


图 10-18　血压数据统计分析

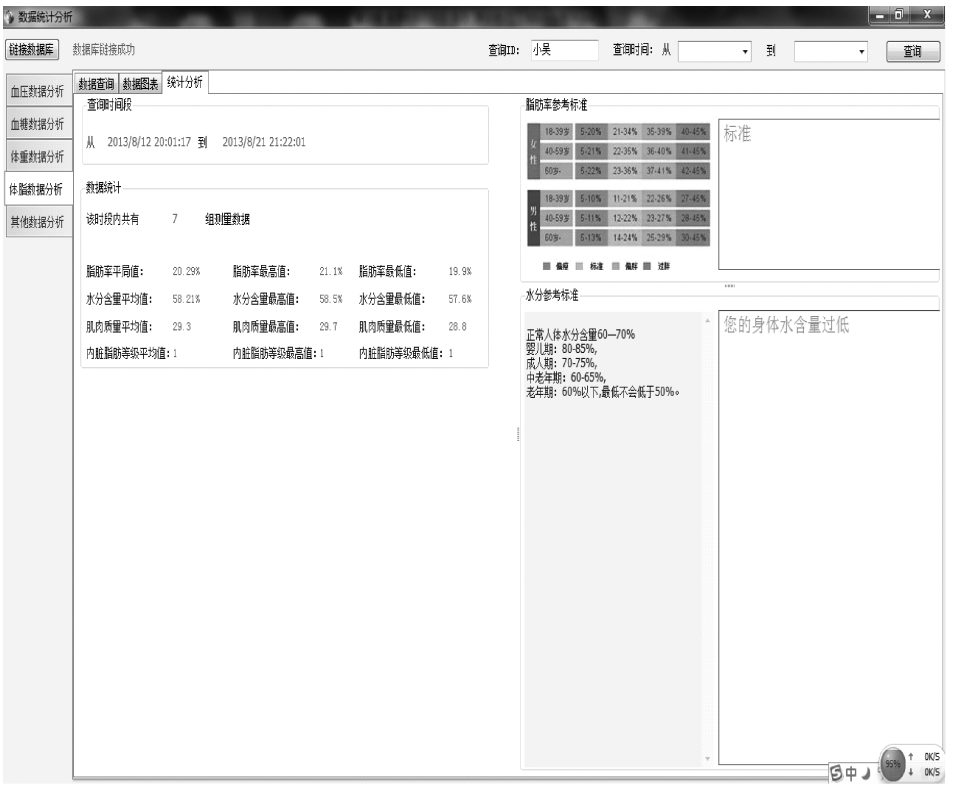


图 10-19　体脂数据统计分析

线。开始测量后，等听到血压计发出“滴滴”声时，再单击“获取数据”，否则可能获取不到数据。

3）当手机或 PC 程序出现错误，请按上述两点检查出错情况。如果还是找不出问题所在，可重新启动程序。

10.7 思考题

1）硬件调试、软件编写过程中遇到的问题都有哪些？分别是怎样解决的？

2）尝试用饼状图反映身体脂肪、肌肉、水分各部分所占的比重。

10.8 实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤（包括软硬件设计方案、软件流程图、硬件原理图等）以及心得体会，最后要附上程序，并按要求回答思考题 1）、2）。

参 考 文 献

[1] Roger S. Pressman. 软件工程：实践者的研究方法（原书第 7 版）[M]. 北京：机械工业出版社，2011.

[2] Scott Millett. ASP. NET 设计模式 [M]. 北京：清华大学出版社，2011.

[3] 陆凌牛. HTML5 与 CSS3 权威指南 [M]. 北京：机械工业出版社，2011.

[4] 国家体育总局. 运动健身指南 [M]. 北京：人民体育出版社，2011.

[5] 尹萍. SQL Sever 数据库性能优化 [J]. 计算机应用与软件，2005，22（3）.

[6] 金华，华进. C#网络编程技术教程 [M]. 北京：人民邮电出版社，2009.

[7] 数据库访问性能优化法则 [EB/OL].

http://www.cnblogs.com/bicabo/archive/2012/09/05/2672709.html，2012.

第 11 章　智能家居网关系统

11.1　实验目的

1）了解智能家居网关组成。
2）学会配置学习型万能红外遥控器。
3）掌握针对学习型万能遥控器的编程。

11.2　实验内容

1）配置学习型万能红外遥控器。
2）使用 Java 语言编写程序。
3）通过使用学习型万能红外遥控器来控制智能家电。

11.3　实验所用仪表及设备

1）硬件：PC 一台、学习型万能红外遥控器一个、载有安卓 2.3.4 系统的智能手机一部。
2）软件：Eclipse 开发环境、安卓 ADT 插件。

11.4　实验原理

11.4.1　智能家居网关系统设计

智能家居网关系统（见图 11-1）以万能学习型红外遥控器和 ARM 架构的智能手机、平板电脑为硬件平台，以安卓操作系统为软件平台，设计与实现了一个具有远程家电控制和虚拟管家语音交互功能的智能家居网关系统。该系统可以实现对家庭设备的语音或手动便捷控制。

系统分为三个部分：远程手持终端、万能遥控器和家庭网关。远程手持终端部分实现复杂网络环境下对家庭设备的远程控制，实现了对智能家居的家庭时况的随时随地地掌控。万能遥控器部分，是针对家庭环境下的具体应用，通过学习型红外遥控器与家庭网络，可以实现家电的便捷控制和远程操作。网关部分，对家庭内部提供与学习型万能红外遥控器的通信功能，对家庭外部，提供网络访问接口，实现远程手持终端的接入。即对内管理家庭内部网络中设备的运作、协调，对外完成设备的远程通信与操作。

系统的技术层次如图 11-2 所示，管理软件运行于安卓设备之上，通过 3G、Wi-Fi 实现对外连接，完成远程和近端的多网络数据交互。在网络接入方面，由于系统提供了 3G 和

Wi-Fi 两条数据通道，因此，可以实现通过 Internet 的访问和通过智能移动终端访问，比如用户在家中时，可以通过手机和平板电脑上的客户端进行家电控制，当用户在公司上班时，可以通过访问服务器得知家庭设备的数据，了解到家中实时状况。用户可以连接学习型万能红外遥控器实现家电控制，同时可以连接监控设备、护理设备等(后续开发中)，从而构成一个完整的智能家居系统。

作为物联网的一个应用，智能家居远程控制系统也具有物联网的典型结构。

1. 家庭网关的设计形式

家庭网关作为连接外部世界与家庭内部世界通信的桥梁，是智能家居物理和逻辑上的核心部件。随着移动互联网、光纤到户等普遍推广应用以及电信运营商的全业务融合，家庭网关已经成为人们数字化家庭的必备产品。目前，家庭网关的实现形式主要有三种，代表着智能家居行业不同的时期。

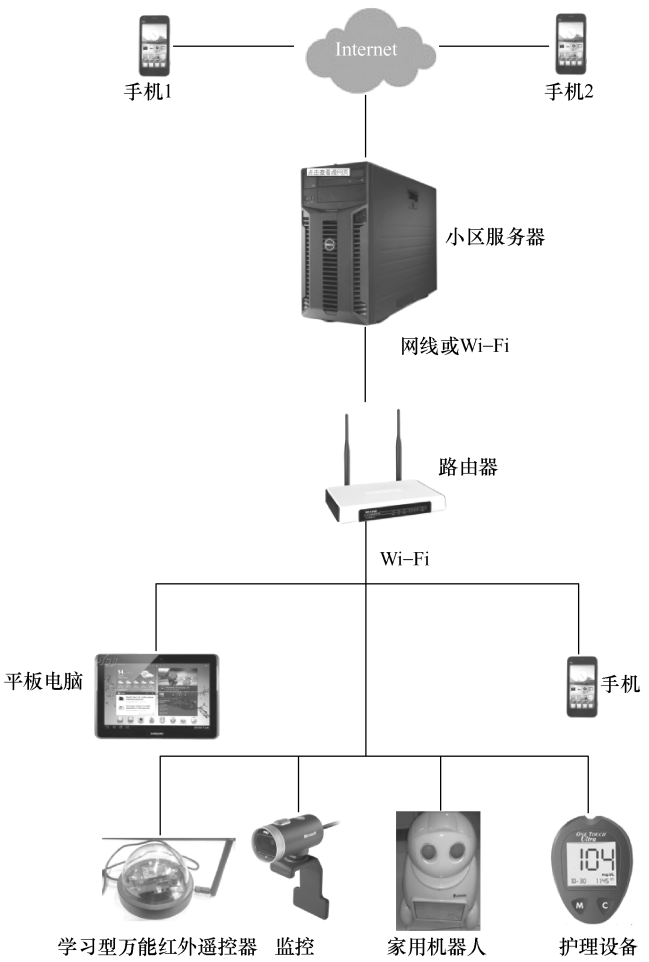


图 11-1 智能家居网关系统示意图

1）基于 PC 架构的家庭网关。该种类型出现于智能家居的萌芽阶段，它将 PC 作为整个

层次划分	实现功能	基本要求	执行器	关键技术
应用层	数据访问和控制命令下达	易交互，实时	服务器/移动客户端	网络编程
网络层	手持设备网络接入	稳定，易实现	3G/Wi-Fi	网络接入
	遥控器网络接入	稳定、延迟低	Wi-Fi	网络接入
感知层	数据采集/家电控制	准确、实时	学习型万能红外遥控器	红外通信

图 11-2 智能家居技术层次图

家庭智能化的控制服务器，加上一些外围的扩展设备组成该系统，实用性很差，易受病毒攻击，基本上只停留在概念阶段。

2）以单片机作为中央处理单元的家庭网关，在实用性和专业性方面提高了许多，但是该控制系统直接对硬件编程，成本高，代码重复利用率低，并且电路复杂，稳定性较差。

3）基于嵌入式系统的家庭网关，对软件进行编辑，灵活应用性强，代码重复利用率高，体积小，功耗低，稳定性强。目前家庭网关设计主要采用这种形式，随着嵌入式技术更加广泛地应用，高性能 CPU、嵌入式操作系统的发展，智能家居正向智能化、人性化、模块化、平民化方向发展。

2. 家庭网关的功能

家庭网关作为一种新型技术产品，具有比以往产品更加丰富的功能与应用，其中，主要围绕家庭网关以下几大功能：

（1）通信功能

家庭网关作为链接外部网络和家庭内部网络的桥梁，能够通过多种方式接入多个网络，在提供接入手段的同时具有服务质量控制能力，保证家庭中不同业务的服务质量，这是家庭网关最基本的功能。

（2）娱乐功能

家庭网关作为一个宽带综合平台，为家庭用户提供各种各样的宽带应用，用户无需借助计算机就可以完成各种宽带娱乐体验以及信息交流和共享，还可以通过提供音乐、视频等数字内容丰富用户的娱乐体验。

（3）节能功能

家庭网关可以通过智能控制家用电器的使用等达到节能的效果，而且家庭网关还是智能电力的一个核心部分，因此节能功能也是其主要功能之一。

（4）与智能家电相关的功能

家庭网关可对连接到家庭网关上的各种智能家电终端进行设备发现和管理，也可通过远程管理功能远程、实时地操控智能家电。

（5）平台功能

家庭网关是一个开放的业务平台，可采用标准接口实现管理、控制以及下载，同时能够融合协调各种不同的业务。

（6）安全功能

家庭网关作为家庭与外部网络的接口必须保证安全性，这种安全性应该是多个层次上的，从信息保护的角度来说，家庭网关具有防火墙功能；从业务安全角度来说，家庭网关可以实现不同业务的安全保护，如上网限制等；同时也能够实现 VPN 功能，建立多个家庭或家庭与办公地点之间的安全虚拟专网。

（7）管理功能

家庭网关的管理功能主要包括：故障管理——家庭网关具有故障管理功能，能够支持远程诊断、故障申报等功能；计费管理——家庭网关综合了家庭多项业务，向用户提供统一的计费功能，屏蔽外部网络的差异，提供统一的按照业务收费的方式；性能、用户管理——家庭网关具有性能管理功能，如显示带宽、流量等功能，同时可以针对家庭中不同的用户设定不同的权限，如儿童限制访问功能等。

3. 本实验设计目标

（1）网络在哪家就在哪

只需要通过智能手机或个人计算机接入平台，就可以使用这些服务来控制家庭设备，如空调、电灯、计算机、电视、音响等。

（2）简洁、高效、廉价的通用平台

管理软件可以运行在各种 ARM 架构的搭载安卓平台的主流手机和平板电脑，使之作为家庭网关。系统提供了 3G 和 Wi-Fi 两条数据通道，并且只关注数据接口而不关注用户设备的形式，很好地屏蔽了不同用户现有设备之间的差异。向下可以连接万能遥控器实现家电控制，系统只操作设备地址，而服务内容和设备操作码可以根据实际应用，在学习型万能红外遥控器中定义，使系统具有更大的灵活性。

11.4.2 智能家居网关的硬件实现

智能家居系统智能化控制的核心是家庭网关，家庭网关是家庭内部网络与 Internet 的唯一通信接口，在系统中起到承上启下的作用，对内综合管理，对外统一接口，是底层设备对应用层的数据转换器，也是应用层对底层设备的命令解释与执行机构。本系统使用的家庭网关在家庭内部，通过一定的传输介质将各种设备和系统连接起来，使这些系统之间可以互相通信，采用统一的通信协议，对内管理家庭内部网中设备的运作和协调；在家庭外部，网关通过 Internet 和 3G 等通信网络实现公众通信平台网络接入，完成设备的远程监控。

家庭网关是智能家居的控制中心，拥有出色的通信与协调能力是家庭网关的重要指标。本系统的多功能家庭网关，实现家庭环境下的家用电器的集中网络化控制，并且具有虚拟管家、语音交互功能，通过云计算识别语音技术，网关实现控制指令的语义识别，实现用语音控制家电的操作。管理软件将原来由遥控器控制的家电集中到平板电脑、手机等终端上，或者通过家庭局域网或 3G 网络进行控制——总体来说，实现了网络化控制，任何有网络的地方，就能随心所欲地控制家庭中的电器设备，在统一平台上实现子系统的服务代理、协议转换和信息处理服务。

图 11-3 学习型万能红外遥控器

本系统的硬件开发工作主要在学习型万能红外遥控器部分（见图 11-3），向上连接家庭无线路由器，向下实现家电控制。

学习型万能红外遥控器的参数和功能：

1）Wi-Fi 无线控制，红外可以全向发射，一个房间只要有一个该设备就可以控制整个房间内嵌 Web 页面（共 30 个按键控制），方便操作，客户利用手机、IPad、PC 等设备通过 Web 界面访问控制设备。

2）可学习市面上所有的电视、空调（特别是日本空调，超长码）、音响、VCD 红外遥控器，适应性强。

3）支持 Wi-Fi 无线组网，支持 AP 模式和 STA 工作站模式。

4）支持 IEEE 802.11b/g/n 无线标准。

5）支持网桥，路由器的网络架构。

6）控制协议完全透明，控制指令简单。

7）提供基于 Internet 的超远程控制的后台软件支持，用户能够在有互联网的地方都能控制设备。

8）可以上传保存学习红外码值，并可以下载学习好的红外码，发送时间 800ms 内完成。

9）红外载波频宽：10～80kHz。

10）内置大容量，本地可记忆 100 个按键指令。

11）PC 端可以无限制存储发射。

12）提供用户任意订制仿真界面的功能，能够设计专门的遥控仿真软件，方便操作。

13）掉电数据不丢失，设置不丢失。

14）提供集控平台软件，解决了智能家居中的遥控设备集中控制难题。

15）控制器提供 Living Update 接口，支持在线更新功能。

16）支持 TCP/IP/UDP 以太网数据接口。

11.4.3　家庭网关的软件平台

智能家居中对家庭智能设备的控制，大多是通过运行在各种 ARM 架构搭载的安卓平台上的软件来实现的。

安卓是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑。安卓的系统架构和其操作系统一样，采用了分层的架构。从架构图看，安卓分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 内核层，如图 11-4 所示。

针对智能家居开发的软件运行于安卓设备之上，使用 Java 语言编写。因此使用流行的 Eclipse IDE 来开发。Eclipse 软件可运行于 Windows、Linux 和 Mac OS 之上。

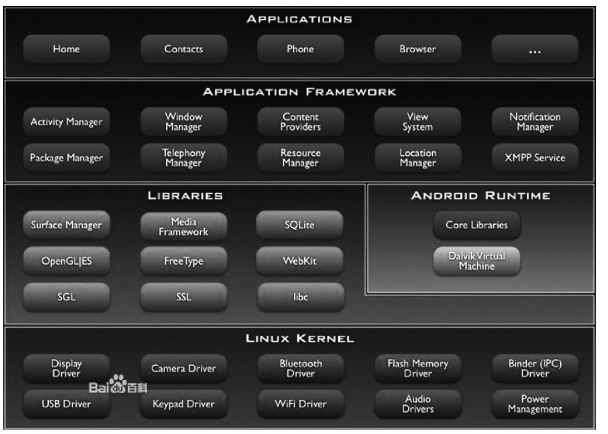


图 11-4　安卓系统结构图

11.5　实验步骤

11.5.1　配置学习型万能红外遥控器

遥控器在 Wi-Fi 通信模式下有两种学习红外操作码的方式。第一种为接入点（Access Point，AP）模式，即本控制器创建 Wi-Fi 网络，供手机、笔记本、平板电脑等其他 Wi-Fi

设备接入，类似于家里的无线路由器，一个无线网络只能有一个 AP。本遥控器最多可以支持 32 路 TCP 链路连接。如果只有一台遥控器同时没有其他无线网络可以利用的情况，可以选择 AP 模式。第二种为 ST（Station）模式，即模块加入 Wi-Fi 路由器创建的 Wi-Fi 网络（也可以是本遥控器 AP 创建的无线网络）。因此，在同一个网络内，可以很方便地进行多设备组网，并操作多台组网设备，通常建议按照 ST 模式工作。

首次使用遥控器，插上天线后，然后用配套的电源给设备供电。设备上电后，电源指示灯会亮。

1）打开电脑的 Wi-Fi 网络，如图 11-5 所示。以 Windows 系统视图为例：设备上电后，大约 10～20s 时间，通过无线的网络列表，会发现 SSID 为“HF-A11x_AP”的无线网络。

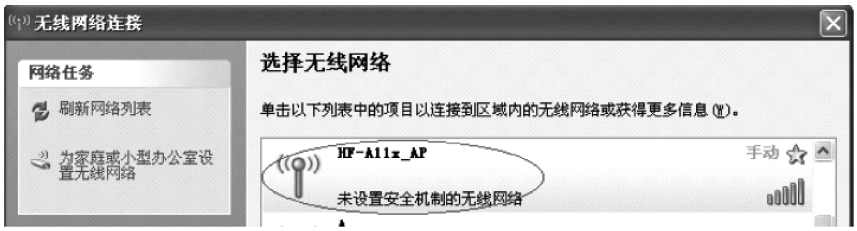


图 11-5　AP 模式下遥控器的 Wi-Fi 网络

加入网络，选择自动获取 IP，Wi-Fi 模块支持 DHCP Server 功能并默认开启，如图 11-6 所示。

图 11-6　连接状态

此时模块的 Link 指示灯亮起。

2）网页配置。

保持 Wi-Fi 网络链接，用浏览器登录 http://10.10.100.254，即可进入设置网页，默认用户名和密码均为 admin，登录界面如图 11-7 所示。

3）登录后界面如图 11-8 所示。

4）设置模块工作模式。

如图 11-9 所示，选择 Station 模式，然后单击“确定”。

5）无线终端设置。

图 11-7　登录界面

图 11-8　登录后界面

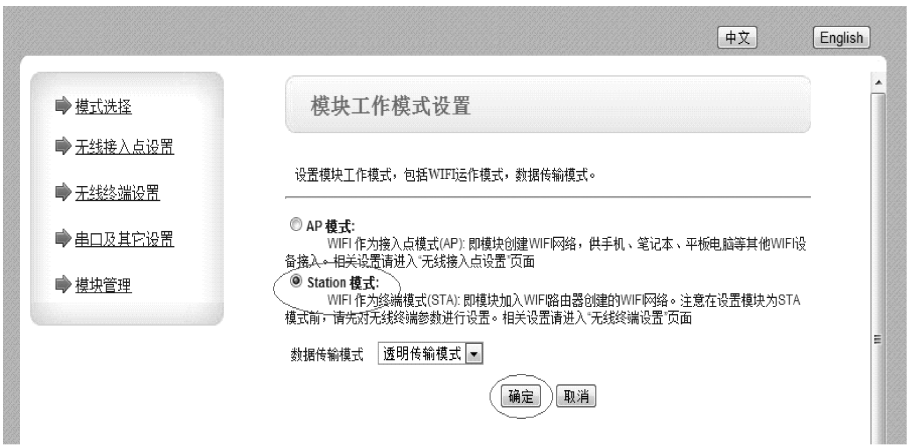


图 11-9　模块工作模式设置

单击无线终端设置里的搜索按钮即可找到网络内的 AP 节点，选择节点，会自动输入加密方式和 SSID 号，用户只需要输入密码即可，如图 11-10 所示。

图 11-10　无线终端参数设置

然后在出现的界面选择需要的路由器，单击“Apply”，如图 11-11 和图 11-12 所示。

Site Survey

	SSID	BSSID	RSSI	Channel	Encryption	Authentication	Network Type
○	1022-II	00:24:a5:ef:ff:e4	0%	1	AES	WPA2PSK	Infrastructure
○	VIDEOPHONE_1UMa	6a:c7:16:2e:d3:11	96%	1	TKIPAES	WPAPSK	Infrastructure
○	STB_1UMa	6a:c7:16:2e:d3:12	96%	1	TKIPAES	WPAPSK	Infrastructure
○	ustb_ustb	1c:fa:68:0c:f7:f6	0%	6	AES	WPA2PSK	Infrastructure
○	Connectify-me	68:17:29:06:61:03	100%	6	AES	WPA2PSK	Infrastructure
◉	TP-LINK_Smarthome418	20:dc:e6:4d:95:1a	86%	6	WEP	OPEN	Infrastructure
○	TP-LINK_AP	ec:17:2f:9a:ab:ba	44%	6	TKIP	WPA2PSK	Infrastructure
○	TP-LINK_king	8c:21:0a:60:c3:12	100%	6	AES	WPA2PSK	Infrastructure

Apply　Refresh

图 11-11　路由器

输入密码：0123456789，然后单击“确定”。

6）串口及网络协议设置。

串口参数设置如图 11-13 所示。

注意波特率为 9600，单击“确定”，网络参数设置如图 11-14 所示。

网络模式为 Server，协议为 TCP，端口号为 5000，然后单击“确定”。

7）配置 IP 地址。

IP 地址：192. 168. 1. 150（静态地址）

子网掩码：255. 255. 255. 0

图 11-12　密码

图 11-13　串口参数设置

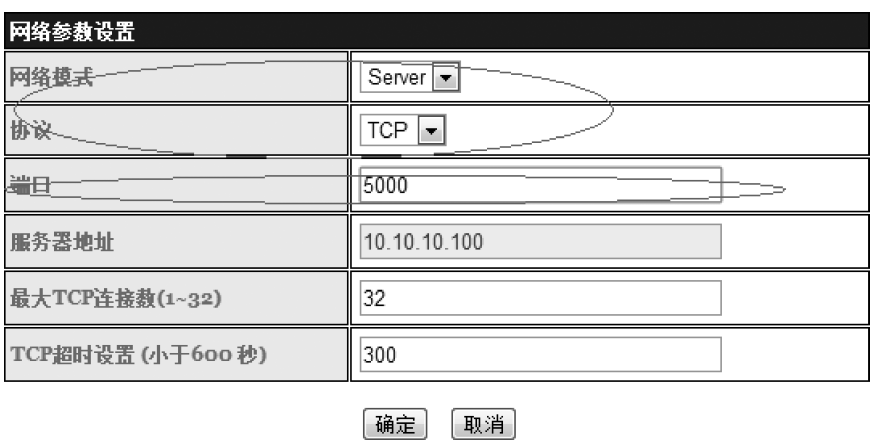


图 11-14　网络参数设置界面

网关设置：192. 168. 1. 1

如图 11-15 所示。

图 11-15　配置静态 IP 地址

8）重启模块。

进入模块管理子页面，重启模块，也可以断电重启，如图 11-16 所示。

图 11-16　重启模块

至此 Wi- Fi 模块设置完成。

11. 5. 2　使用手机控制遥控器学习红外信号

1. 安装安卓软件

安装包为 smarthome4cell. apk，安装到手机（使用 360 手机助手等工具，或复制到手机内存卡中），本软件正常运行需要手机有内存卡。安装完成后的界面如图 11-17 所示。

2. 学习红外——设置

在主界面单击“控制界面”按钮，进入功能列表，如图 11-18 所示。

选择“系统设置”选项，进入设置界面，如图 11-19 所示。按照图示参数进行设定，然后单击“确定”按钮，即可完成设置。

3. 学习红外——学习

在“功能列表”菜单界面选择“家电控制”选择，单击进入，界面如图 11-20 所示。选择想要学习的电器，进入相应的界面，如图 11-21 所示。

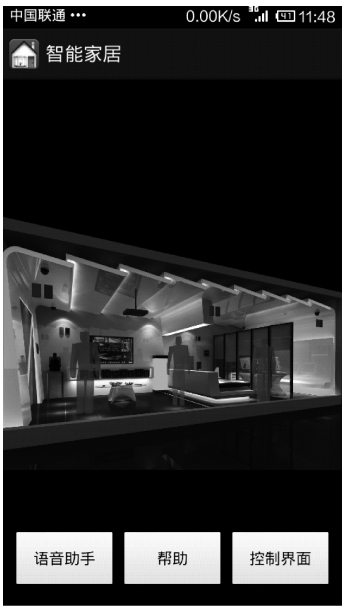


图 11-17　安卓软件主界面

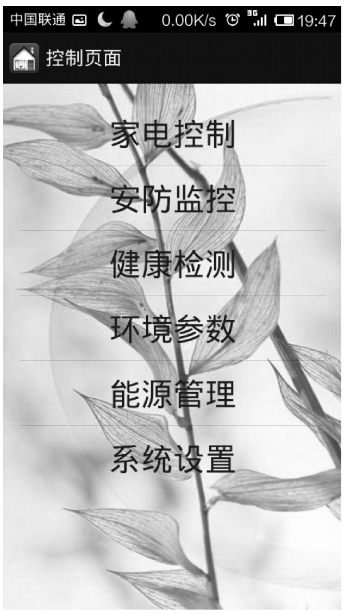


图 11-18　功能列表

图 11-19　系统设置

图 11-20　家电控制页面

单击遥控器相应的按钮，此时会看到万能遥控器上的绿色指示灯熄灭，表示万能遥控器进入学习状态，这时再将电器遥控器对准万能遥控器上的红外学习头，如图 11-22 所示。单击对应按钮进行学习，学习完成之后，绿灯点亮。学习完所有按钮之后，再回到“系统设置”界面，关闭学习模式，就可以进行家电的控制了。

控制器自带有 100 个红外码的存储空间，即使断电，已经学习到的红外码依然存在。通常将经常使用的红外码存储到本空间。

图 11-21 空调、电视遥控界面

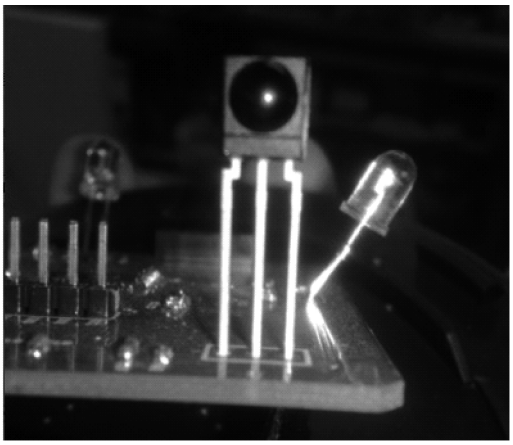


图 11-22 红外学习头

11.5.3 在 Eclipse 中模拟学习型万能红外遥控器网关指令

本实验是在 Eclipse 环境中，使用 Java 语言针对学习型万能红外遥控器设计的安卓虚拟程序。该程序通过模拟界面输入遥控器控制指令来实现对家电的控制，所输入的指令就是红外学习之后存储在遥控器本地的家电地址。遥控器通过对应用程序中指令代码的判断来发起网络连接，这也就实现了网关的功能。

1. 学习遥控器指令操作码

控制遥控器的十六进制指令格式如表 11-1 所示，其中指令地址即为存储在遥控器本地的红外操作码的存储地址。

表 11-1 遥控器指令格式

长度	类型	发送码	指令地址	预留	后三位的异或和
50	fa	01	XX	0	XX

程序中指令发送关键代码如下：

```
SendCode1[0] = 0x50;//
SendCode1[1] = (byte)0xfa;//强制转型为字节型以便发送
SendCode1[2] = 01;
SendCode1[3] = instruction;//instruction 为 int 型变量,为 0 ~ 100 之间的十进制数,表示红外操作码的存储地址
SendCode1[4] = 0;
SendCode1[5] = (byte)(SendCode1[2]^SendCode1[3]^SendCode1[4]);
try {

Socket sck = new Socket(strIp,5000);//发送指令到遥控器的指定端口,strIp 为预先定义的遥控器的 IP 地址,端口号为 5000
    sck. setSoTimeout(3000);
```

```
        InputStream in = sck. getInputStream( ) ;
        OutputStream out = sck. getOutputStream( ) ;
        out. write( SendCode1 ) ;
        out. flush( ) ;
        sck. close( ) ;
    }
```

表 11-1 中的指令地址，也就是程序代码中的 SendCode1［3］，就是遥控器所要控制的家电地址，其为我们之前进行红外学习时选择的红外码存储空间地址。图 11-23 和图 11-24 为已经学习后的红外空间地址。

1	电视开		设置：	接入网络	TP-LINK_S marthome418							
2	关			IP 地址	192.168.1.150		(注意！IP 设置成150之后，同一时刻只能有一个万遥接入网络！)					
3	频道+			加密	WEP	64位	16进制	密钥：0123456789				
4	频道-											
5	音量+											
6	音量-											
7	主页											
8	返回											
9	确定											
10	箭头上											
11	箭头下											
12	箭头左											
13	右											
14	空调开											
15	空调关											
16	温度+（30度）											
17	温度-（16度）											
18	制冷											
19	上下扫风											
20	左右扫风											
21	电灯1开											
22	电灯1关											
23	电灯2开											
24	电灯2关											
25												
26												
27												
28												
29												
30	电视静音											
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41	空调制热											
42												
43												
44												
45												
46												
47												
48												
49												
50												

图 11-23　红外学习后的指令地址列表

51														
52														
53														
54														
55														
56														
57														
58														
59														
60														
61														
62														
63														
64														
65														
66														
67														
68														
69														
70														
71														
72														
73														
74														
75														
76														
77														
78														
79														
80														
81														
82														
83														
84														
85														
86														
87														
88														
89														
90														
91														
92														
93														
94														
95														
96	空气净化器开													
97	空气净化器开													
98	空气净化器开													
99														
100														

图 11-24　红外学习后的指令地址列表

2. 模拟学习型万能红外遥控器网关指令

在 PC 运行已经安装好的 Eclipse 软件，打开配置好的工程后便会出现如图 11-25 所示网关测试界面。根据学习型万能红外遥控器的指令格式逐个填写，单击“发送”后将会正确控制智能家居中的家电。因为我们是在 PC 中虚拟出的界面，实际操作中，指令不会发送到家电。如果把该测试程序安装在安卓手机上，接入智能家居网络后将会正确地控制家电。

图 11-25　网关测试界面

11.6　思考题

1）家庭网关应该实现哪些功能？

2）对学习型万能红外遥控器进行配置时，是如何做到让遥控器判断并正确控制家电的？

3）程序中，是通过哪些代码来完成对家电指令的正确发送和控制的？

11.7　实验报告内容及要求

实验报告内容应包括设计目的、设计内容、设计所用仪表及设备、设计原理、设计步骤（包括软硬件设计方案、软件流程图、硬件原理图等）以及调试过程、心得体会，最后要附上程序，并按要求回答思考题 1）、2）、3）。

参 考 文 献

[1] 王志良，王粉花. 物联网工程概论［M］. 北京：机械工业出版社，2011.

[2] 王志良. 物联网现在与未来［M］. 北京：机械工业出版社，2010.

[3] 李界家. 智能建筑办公网络与通信技术［M］. 北京：清华大学出版社，2004.

[4] 张云. 基于 ARM 的智能家居远程监控系统研究与实现［D］. 阜新：辽宁工程技术大学，2009.

[5] 杨晨. 嵌入式智能家居控制系统的研究［D］. 哈尔滨：哈尔滨理工大学，2007.

[6] 敖志刚. 智能家庭网络及其控制技术［M］. 北京：人民邮电出版社，2011.

[7] 谭朋柳，舒坚，吴振华. 一种信息-物理融合系统体系结构［J］. 计算机研究与发展，2010，S2.

[8] 沈苏彬，范曲立，宗平，等. 物联网的体系结构与相关技术研究［J］. 南京邮电大学学报：自然科学版，2009，29（6）.

[9] 王志良，王鲁. 物联网终端技术［M］. 北京：机械工业出版社，2013.

第 12 章　智能家居综合实训学生心得

学生心得（1）

总体来说，本次物联网实训课程是非常有意义的。老师常说“人”这个字这一撇和一捺分别代表着“知识”和“技能”，只有两样都掌握好才能立足，确实在大学阶段我们主要学习的还是“知识”，而“技能”方面的实践却比较少。不过自从分到物联网专业以来，老师一直强调要注重实践，特别是本次的物联网综合实训，使我们接触到了一个完整的物联网应用——“智能家居系统”，这个系统正是物联网在生活中的具体体现。

在平时的学习过程中，我们学到的大都是书本上设计好的场景，而对于物联网是否真正地走进了生活不甚了解，始终觉得如镜中月、水中花一般。经常被其他专业的同学问起物联网究竟是什么，到底能干些什么。按照以前我会把平时课上学到的说给他们听，但总是缺乏真实性。不能使人信服。经过这次的实训，使我对物联网的应用有了更深刻的认识，再向他人解释的时候，我相信自己能更好地说明物联网的应用。

在智能家居系统的实验室中，我们看到了平时接触到的物联网的各种器件能真正地综合成为一个系统，为人们的生活提供便利，使生活更舒适。

我相信在学校中没有其他任何专业能像物联网专业这样能真正接触到实际的应用，这也使我明白此次的实训来之不易，更应该珍惜。也明确了自己的责任，要更加注重自己的“技能”学习，在下学期我要多编程，多实践，为物联网的发展尽自己的一份绵薄之力。

学生心得（2）

这门实训课是老师特意为我们首届物联网班的学生添加的，考虑到能够让每名同学都亲自参与实训，将我们分为 3 人一组做不同的内容，与多个博士生交流学习，慢慢接触了解了一些科研项目，在多方面的帮助下学习到了许多知识。

就实训内容而言，收获良多，从课件上看到的一些设备能够亲眼见到，并且使用它，在老师的引导下开始思考学物联网这门专业的知识是为了要实际应用，要真真实实地做出些实用的东西方便大家，例如：健康设备，就给我们提供了很好的实验平台。构想自己身在外面，能通过手机了解到家里发生的一切，能通过手机控制家里的电器运行，窗帘拉开，烧水等，这些无不需要物联网的强大支持才能实现，更加确立了物联网的方向。

学生心得（3）

打开家居实验室的门，就给人眼前一亮的感觉，真是一个巨大的惊喜。之前对智能家居的概念，都只是停留在理论的阶段，最多通过图片或者视频，有一个视觉上的初步认识，而在实验室中，能给人的各项感官一个全方位的刺激。

介绍了各种设备，都让人惊奇。首先最震撼的就是云电视，感受到科技的进步——不仅是底层的架构如：布线、网络；或是基础的设施如：传感器、监控；还是应用设备如：各种健康测量仪器——处处都是智慧的体现。

给人找到了学习方向的感觉，之前不知道学的东西都有什么用，而现在实际是看到了用处。例如已经申请了专利的万能遥控器，它的设计想法就非常好，到最后实现的功能也很强

大，特别是它的学习功能，增强了交互性与可扩展性。

学生心得（4）

本次综合实训的课程的重点在于锻炼我们这批物联网专业学生的专业知识的学习和动手实践的能力。正如老师反复强调的那样，物联网作为 21 世纪一种建立在 IT 领域各项高级发明的基础上，融合了各个学科的专业知识的综合型专业，所要求的人才不仅仅需要在知识层面上融会贯通，还应该在专业技能上高人一筹。为此，系里这次在人力、物力等各个方面都投入了很大的精力，筹备搭建而成智能家居实验室。

实验室雏形搭建基本完成之后，我们班有幸作为第一批使用者亲眼见到了老师和学长们口中所说的智能家居大系统。不得不承认这短短的一次实训课程比平时上课要直观有效得多，之前脑中对于智能家居的概念还仅仅停留在理论知识阶段，都是通过文字、图片、视频等途径获取到的，是别人“给”的知识，是被动接收的信息。而这次通过老师和学长们的介绍和演示，当我们站在真实的场景中看到各种实物时，会不自觉地产生疑问，从而去寻求解答，这样主动得来的信息是“取”的知识，更容易被大脑记牢。这次的实训过程中，平时接触到的书本上介绍的各种枯燥的物联网名词好像都变得生动鲜活起来。

学生心得（5）

物联网实训课的课程安排内容围绕着正在建设中的智能家居实验室技术环节而展开，从物联网应用体系结构由浅入深地讲解了一些基础技术知识。课程内容安排考虑到了让物联网专业的学生广泛并且较为全面地去了解与体会物联网技术架构中的各个分支，能够从总体上指导学生进入物联网的具体应用世界，尤其是在智能家居方面的应用。

之前对于物联网并没有系统地去接触一些在生活中的应用场景与产品，对物联网的分支技术要求、难题以及应用带来的优势与变化并没有具体、直观的认识。这门课为我们学生提供了这个条件，尤其是针对家居实验室建设过程中的实际支撑技术做了相关的介绍以及鼓励学生自己编写应用联入开放式的智能家居系统当中。在实践中应用理论知识，感受物联网产品的实际价值对于我们第一届甚至是后面的物联网专业的学生有很好的指导与鼓励作用。

学生心得（6）

这次的实训经历了将近一个学期的时间，听老师和学长们讲了很多关于智能家居的技术，也终于参观了实验室。分组学习是一个好的方式，我们首先了解到自己感兴趣的那部分内容在整个智能家居系统里充当着怎样的角色，可以用它来完成哪些功能，也开拓了我们的思维。每人一个程序也“逼”着我们动手去做点什么。没有任何内容上的限制，只为实现一个点子而去编写程序，这才是程序员写程序的乐趣所在。当然在过程中，我们能学到很多知识。兴趣是最好的老师。

总之，这次实训让我们开了眼界。先进的设备和先进的技术结合出的全新的理念，让我们对未来有了更多憧憬与希望，看到了方向所在，让我们更有动力去学习，实践，一步步实现自己脑海里的未来。

学生心得（7）

通过本课程的学习，我深入了解了物联网在当今的应用趋势，感受到了智能化所带来的革新，以及我们生活与物联网的息息相关，让我对继续深入研究物联网的信念更加坚定。

首先，物联网体系结构课程中，通过老师和学长的讲解和介绍，让我们对于物联网的智能应用和具体实现有了初步的掌握，认识到了学习任何一个学科，首先要做的便是踏踏实实

地学习这一学科的基本技能和研究方法，在熟练掌握这一技能之前，任何设想都只能是空想，因为我们还没有真正的实践能力，还没有进行将理想转化为现实的实践，所以，唯有好好学习基础技能，掌握正确而独特的研究方法，才能实现自己的专业目标，从而有所建树。

其次，在我们进行物联网智能家居实验室的创构时，根据自己的意向选组，进而由不同的学长带领我们去学习既定的知识，从而实现了实验室整体构想中的“部分零件”的学习和掌握，并且，在老师和学长的帮助下，我们由本科生的懵懂，开始步入到研究生的“研究课题”之中，这一步看似简单的跨越，实际上是让我们提前感受到了研究生的学习生活和奋斗过程以及最终的目标，让我们追逐研究生生活的意识更趋强烈，对于我们这些考研的同学来说，这无疑是精神上的莫大鼓舞，让我们在繁重的复习之中找到信念的坚定力量。

最后，通过本课程，我树立了一个个人目标——充分发挥自己的绵薄之力以帮助推动物联网的普适化，不断发展物联网使之真正地应用于各家各户！

参观了物联网智能家居的实验室，我很兴奋，同时又对这个专业更加抱有希望——我们终于能看到实实在在的物联网的应用，它的确可以如此普适地融于我们的生活，而不是像从前的种种设想那样只能天马行空终究却是梦一场。

学生心得（8）

在参观我们系的智能家居样板间时，一进门，我就被各种“高科技”所吸引，看样子未来如果我们都能有这样的家居环境，那么一定会更加享受生活带来的方便。等到老师给我们一样样讲解，我突然发现这些不都是我们课上所学的东西么，原来感到那么遥远的知识，却如此近在眼前，很现实。之前感觉没学到真本领而有些沮丧，但现在看到这些实物很受鼓舞，物联网是很有发展前途的事业，我们的发展空间很大。

学生心得（9）

自从选择了物联网这个专业以来，我已经学习了很多专业课，但还没有见过物联网的实物，它对我们还是一个概念。而通过物联网综合实训这门课程，我们不仅对物联网各方面进行了学习，更加切实看到了物联网的实际应用，体会到了它的先进之处。

本周我们进入了智能家居实验室进行实训，实验室的建设使我们感到惊奇，不禁感叹。在我们的概念里，智能家居系统这个词离我们的生活似乎异常遥远，只是存在于高科技展览会里的表演，或是富豪们向人们炫耀财富而展示的一种家居装饰。事实上，近年来智能家居系统已开始一步一步走进越来越多的普通家庭。一套完整的智能系统一般包括安防、电话、影音、空调、灯光等子系统，价格上已可以被消费者所接受，完全可以根据自己的需要和喜好选择安装子系统，以满足自己居家的各种需求。所以它已成为一种趋势，也成为了我们的研究目标。

老师说，学习技术不是枯燥的，而是一件快乐的事，它可以把你的想法变成现实，现在我们更加深刻地体会到了这句话。是我们对物联网行业发展的前景充满信心，也激励我们不断努力学习，积极创造，以后为推动整个物联网行业的发展做出贡献。

地址：北京市百万庄大街22号
邮政编码：100037
电话服务
社服务中心：010-88361066
销售一部：010-68326294
销售二部：010-88379649
读者购书热线：010-88379203
网络服务
教材网：http://www.cmpedu.com
机工官网：http://www.cmpbook.com
机工官博：http://weibo.com/cmp1952
封面无防伪标均为盗版

上架指导： 工业技术／物联网
ISBN 978-7-111-45808-1
ISBN 978-7-89405-282-7（光盘）
策划编辑◎朱 林/ 封面设计◎赵颖喆

ISBN 978-7-111-45808-1
9 787111 458081 >
定价：48.00元（含1DVD）